UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

João Carlos Redin

AVALIAÇÃO DO IMPACTO NAS MÉTRICAS DE DESEMPENHO DE PROTOCOLOS DE ROTEAMENTO EM REDES MANET.

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Prof. Dr. Carlos Becker Westphall

Florianópolis, Fevereiro de 2005

AVALIAÇÃO DO IMPACTO NAS MÉTRICAS DE DESEMPENHO DE PROTOCOLOS DE ROTEAMENTO EM REDES MANET.

João Carlos Redin

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, Área de Concentração **Sistemas de Computação** e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Prof. Dr. Carlos Becker Westphall

Orientador

Prof. Dr. Raul Sidnei Wazlawick

Coordenador do Programa de Pós-Graduação em Ciência da Computação

Banca Examinadora

Prof. Dr. Carlos Becker Westphall
Presidente

Prof. Dra. Carla Merkle Westphall

UFSC

Prof. PhD Mário Antonio Ribeiro Dantas

UFSC

Prof. Dr. Roberto Willrich

UFSC

DEDICATÓRIA

A Deus, pelas lições que me tem proporcionado.

AGRADECIMENTOS

Ao meu Orientador, Professor Carlos Becker Westphall, pelo apoio, compreensão, paciência, inteligência e dedicação.

À minha Mãe, Dolori Redin, que, além de me dar à vida, me ensinou a ter garra, ser uma pessoa honesta e, acima de tudo, sempre buscar fazer a minha parte para viver em mundo melhor.

À minha querida irmã, Salete, que me ajudou a viabilizar o presente Curso de Mestrado.

À minha esposa, Márcia, que me incentivou quando eu desanimava e me compreendeu nas minhas ausências.

À minha equipe de trabalho, que muitas vezes entendeu quando o stress me abatia.

Aos meus amigos, que estiveram juntos nesta caminhada e a todas as pessoas que de alguma forma contribuíram para a realização deste trabalho e foram solidários nas mais diversas situações.

A Deus, por haver me dado forças, mostrando-me sempre novos caminhos.

Sumário

Lista de Figuras

Lista de Abreviaturas

ACK	*Acknowledgment*
AIFS	*Arbitration Interframe Space*
AP	*Access Point*
BA	*Basic Access*
BSA	*Basic Service Area*
BSS	*Basic Service Set*
CA	*Positive Acknowledge*
CBR	*Constant Bit Rate*
CSMA/CA	*Carrier Sense Multiple Access with Collision Avoidance*
CTS	*Clear to Send*
CW	*Contention Window*
DARPA	*United States Defense Advanced Research Projects Agency*
DBPSK	*Differential Binary Phase Shift Keying*
DCF	*Distributed Coordination Function*
DFS	*Dynamic Frequency Selection*
DIFS	*Distributed Interframe Space*
DQPSK	*Differential Quadrature Phase Shift Keying*
DS	*Distribuition System*
DSSS	*Direct Sequence Spread Spectrum*
EDCF	*Enhanced Distributed Coordination Function*
ESS	*Extended Service Set*
ESM	Estação de Suporte a Mobilidade
FCC	*Federal Communications Commission*
FCS	*Frame Check Sequence*
FHSS	*Frequency Hopping Spread Spectrum*
FIFO	*First In, Frist Out*
FM	Modulação de Freqüência
GFSK	*Gaussian frequency Spread Spectrum*
HC	*Hybrid Coordinator*
HCF	*Hybrid Coordination Function*
HiperLAN	*High Performance Radio Lan*
HomeRF	*Home Radio Frequency*
iBSS	*Independence Basic Service Set*
IEEE	*Institute of Electrical and Electronics Engineers*
IETF	*Internet Engineering Task Force*
IFS	*Inter Frame Space*
IR	Raios Infravermelhos
ISM	*Industrial, Scientific and Medical*
ISO	*International Organization for Standardization*
LAN	*Local Area Network*
LLC	*Logic Link Control*
LMDS	*Local Multipoint Distribution System*
MAC	*Medium Access Control*
MAN	*Metropolitan Area Network*
MANET	Mobile *Ad Hoc* Network
MRL	Tabela de Lista de Retransmissão de Mensagem
MPR	*Multipoint Relays*
NAV	*Net Allocation Vectors*
NM	Nodo Móvel

NS	*Network Simulator*
OSI	*Open System Interconnection*
OFDM	*Orthogonal Frequency Division Multiplexing*
ORA	Abordagem de roteamento ótimo
PA	Ponto de Acesso
PAN	*Personal Area Network*
PCF	*Point Coordination Function*
PCM	*Pulse Code Modulation*
PF	Fator de Prioridade
PHY	*Physical Layer*
PN	*Pseudo Noise*
PRNET	*Packet Radio Network*
PPM	*Pulse Position Modulation*
PS	Provedor de Serviço
RF	Rádio Freqüência
RFC	*Request For Comments*
RPF	*Reverse-path Forwarding*
SA	Disponibilidade do Serviço
SAP	Service Access Point
SE	Elemento de Serviço
SIFS	*Short Interframe Space*
SLA	*Service Level Agreement*
STA	Estações 802.11
SURAN	*Survivable Adaptive Network*
TC	*Traffic Categories*
TCP/IP	*Transmission Control Protocol/Internet Protocol*
TI	*Tecnologia de Informação*
TPS	*Transmit Power Control*
TxOp	*Transmission Opportunity*
UDP	*User Datagram Protocol*
UNII	*Unlicensed National Information Infrastructure*
VoIP	*Voice over IP*
WAN	*Wide Area Network*
WAP	*Wireless Application Protocol*
WGs	*Working Groups*
WEP	*Wired Equivalent Privacy*
WI-FI	*Wireless Fidelity*
WLAN	*Wireless Local Area Network*
WLL	*Wireless Local Loop*
WPAN	*Wireless Personal Area Network*
WirelessMAN	*Broadband Wireless Metropolitan Area Network*
VSN	Vetor de Seqüência de Números

Resumo

As promessas das redes móveis Wireless em resolver problemas do mundo virtual continuam chamando a atenção dos projetos de pesquisas industriais e acadêmicos. O aumento da performance dos computadores e da tecnologia de comunicação Wireless é notável. Também se evidencia a necessidade de mobilidade e conectividade que o mundo moderno impõe.

A maior parte das pesquisas sobre redes Wireless foi realizada para o desenvolvimento dos mecanismos básicos de operação. Nos últimos tempos, contudo, foco que tem chamado muito a atenção tange os aspectos de segurança. Considerando que este tipo de redes tem sido idealizado para operar em ambientes hostis, onde os sinais possam ser interceptados e/ou alterados, como as aplicações comerciais e as militares, os mecanismos de segurança devem variar de acordo com o tipo de aplicação a que se destinarem.

Esta dissertação tem por objetivo efetuar um mapeamento dos diferentes tipos de protocolos de roteamento para redes MANET e, a partir daí, pretende-se realizar simulações no Network Simulator avaliando a latência, jitter, vazão e perda de pacotes, checando, assim, o desempenho dos protocolos que implementam conceitos mais sólidos de segurança, comparando-os com protocolos que não possuam esta característica.

Cada um dos protocolos tem suas vantagens e desvantagens e, pela própria característica da rede, a escolha do protocolo que melhor se adeque as características de mobilidade e volatilidade de rede é uma tarefa complexa. As informações das simulações podem contribuir para se traçar um perfil dos possíveis impactos causados com a agregação de segurança aos protocolos aqui implementados. Os resultados apresentados nas simulações desta dissertação mostram que os protocolos seguros de melhor desempenho e que mais atendem aos requisitos de segurança são o Ariadne e o SEAD.

Palavras chave: Segurança, Redes *Ad Hoc Wireless*, protocolos de roteamento, IEEE 802.11, *Ad Hoc*, MANET.

Abstract

Wireless networks promises to solve world virtual problems are grabbing attention from academicals and research projects. In fact, computer performance and Wireless network are rising, following our mobile and connectivity necessities.

Most of Wireless network researchs were done focusing in the development of basic operating mechanism. However, nowadays the security aspects are in evidence. Considering that this kind of network has been built to work in hostile environments as commercial and military applications, the security mechanisms must vary accordingly to the application usage.

The main objective from this document is to map the different types of routing protocols for MANET networks and, after this, to realize tests at Network Simulator to simulate latency, outflow and lost of packs, in order to check the performance of some protocols that are using solid security concepts and others without such technology.

Each protocol has distinct vantages and disadvantages and, based on networks characteristics, to choose the protocol that fits best mobile and volatility necessities is a tough decision. Collected information from simulations may contribute to determine a profile of the possible impacts due to implemented security mechanisms.

Collected results from the simulations realized in this work show us that Ariadne and SEAD are doubtless secure protocols that have best performance, fulfilling all required security aspects.

1. Introdução

Na última década, tem-se observado um grande crescimento nas áreas de comunicação celular, redes locais sem-fio e serviços via satélite. Manifesta-se, também, uma grande diversidade de dispositivos móveis, tais como *palmtops*, *handhalds* e *notebooks*. Com isto, cada vez mais tem se sedimentado o paradigma da computação móvel, advinda de toda esta tecnologia de rede sem fio e dos sistemas distribuídos.

Para suportar a amplitude de aplicações e coberturas, surgiram, nos últimos anos, diferentes padrões. Protocolos e novas tecnologias estão sendo implementados, especificamente, para suprir as necessidades dos ambientes que utilizam dispositivo de comunicação *Wireless* (sem fio). Devido ao crescimento deste segmento de computadores pessoais portáteis, estima-se que, em poucos anos, será bastante comum às pessoas possuírem algum tipo de dispositivo portátil com capacidade de se comunicar com a parte fixa da rede e até com outros dispositivos móveis.

Para prover mobilidade e tornar as tecnologias escaláveis, foram necessárias as concepções de padrões para as redes de telefonia celular, para as WLAN (*Wireless Local Area Network*), Wi-Fi (*Wireless Fidelity*), *Hiperlan* (*High Performance Radio Lan*), utilizados em ambientes locais; padrões para as Wpans (*Wireless Personal Area Network*), empregadas na comunicação de redes pessoais; e padrões para os sistemas móveis, representados pelo protocolo WAP (*Wireless Application Protocol*) [Silva, 2004].

Apostando no crescimento das redes de comunicação sem fio, instituições e empresas investiram em tecnologia, de forma que a mesma pudesse ser utilizada não somente em redes de longas distâncias, mas também, em redes locais, como é o caso das WLANs. Esta tendência foi fortalecida pelo IEEE (*Institute of Electrical and Electronics Engineers*) [Silva, 2004].

Inicialmente, o IEEE constituiu um grupo de pesquisas para criar padrões abertos, que pudessem tornar a tecnologia *Wireless* cada vez mais realidade. Muitos esforços foram empregados no sentido de fazer com que a tecnologia sem fio fosse de fato aceita. O projeto chamado de Padrão IEEE 802.11) [IEEE, 1999] permaneceu, contudo, na inércia por alguns anos.

Um dos fatores que mais interferiu no desenvolvimento tecnológico foi à baixa taxa de transferência de dados, que, inicialmente, contemplava apenas alguns Kbps. No instante em que a velocidade de transmissão se elevou para 1 a 2 Mbps, a rede sem fio passou a ser vista como uma tecnologia promissora e, a receber incentivos para a construção de equipamentos que permitissem a interoperabilidade entre dispositivos de diferentes fornecedores.

Convém anotar que, em geral, os ambientes sem fio são mais inseguros que as redes cabeadas. Assim, permitir que um usuário se desloque entre diferentes áreas, sem a perda de conectividade, pode representar um problema em potencial, uma excelente porta para invasões, se não forem observados certos critérios de segurança.

A maior dúvida sobre o uso de redes sem fio recai, portanto, sobre o fator segurança. Com um transmissor irradiando os dados transmitidos através da rede em todas as direções, como impedir que qualquer um possa se conectar a ela e roubar seus dados? Um ponto de acesso instalado próximo à janela da sala provavelmente permitirá que um vizinho, a dois quarteirões da sua casa, consiga captar o sinal da sua rede.

A preocupação com o fator segurança fica agravada pela popularidade de uso que as redes sem fio vêm ganhando. Este sucesso é devido à característica intrínseca das redes *Ad Hoc* de não possuir fios, o que facilita a instalação, extensão e manutenção de uma rede e, principalmente, permite maior mobilidade. Toda uma nova geração de dispositivos portáteis vem sendo disponibilizada comercialmente, tais como telefones celulares, assistentes pessoais digitais (*Personal Digital Assistant* - PDA) etc.

Por outro lado, a comunicação sem fio é normalmente realizada através de radio freqüências, que é um meio altamente vulnerável, uma vez que permite a escuta da comunicação com o uso de qualquer dispositivo que esteja no raio de ação do transmissor. Assim, a comunicação sem fio *Ad Hoc* precisa ser acompanhada de mecanismos de segurança, a fim de garantir a integridade e a privacidade da comunicação, bem como a autenticação das entidades envolvidas. Entretanto, este é o ponto fraco deste tipo de redes e, certamente, foco de muitas pesquisas.

O controle de acesso e a confidencialidade precisam ser garantidos mais do que nunca, sendo que, em geral, as redes *Wireless* não garantem qualquer segurança, quando usadas na sua forma mais simples.

Formalmente, os níveis de segurança podem ser definidos com um conjunto de requisitos no qual exige-se que determinados itens sejam observados, como o uso da criptografia no armazenamento das informações, a utilização de *firewalls* nos pontos de acesso etc. Enfim, a segurança das informações é e continuará sendo fundamental, seja utilizando redes sem fio ou, até mesmo, redes convencionais.

Esta possibilidade de segurança dará alguma mobilidade para o usuário levar o notebook para onde quiser, dentro de casa, sem perder o acesso a Web, sendo ainda mais interessante seu uso para empresas e escolas.

No caso das empresas, a rede móvel *Ad Hoc,* também conhecida como *MANET* (*Mobile Ad Hoc Networt),* permitiria que os funcionários se deslocassem pela empresa, sem perder a conectividade com a rede. Bastaria, por exemplo, ao funcionário entrar pela porta para que o seu notebook automaticamente se conectasse à rede e sincronizasse os dados necessários.

Tem sido verifica que, no caso das escolas, a principal utilidade das redes *MANET* seria fornecer acesso a Web aos alunos. Esta já é uma realidade em algumas universidades.

Neste trabalho, iremos efetuar um mapeamento dos diferentes tipos de protocolos de roteamento para redes MANET e, a partir daí, pretende-se realizar simulações no Network Simulator, avaliando o desempenho dos protolocos, checando, com isto, o impacto dos protocolos que implementam conceitos mais sólidos de segurança, comparando-os com protocolos que não possuam esta característica.

1.1. Objetivos

O objetivo geral deste trabalho é estudar os protocolos que dão suporte às redes *MANET*, conhecer as suas vulnerabilidades, as técnicas de ataques para o comprometimento destas redes, buscando fornecer uma análise dos desempenhos obtidos pelos protocolos seguros comparando-os com os que não implementam estas características.

Ao término da dissertação, almeja-se que os objetivos específicos listados a seguir sejam alcançados:

Proporcionar um material teórico, experimental e prático a respeito do tema: análise de desempenho de redes de comunicação sem fio *MANET*.

Realizar simulações nas redes MANET para obter informações comparativas de desempenho, a fim de avaliar os possíveis impactos causados com as implementações de requisitos de segurança aos protocolos de roteamento.

1.2. Justificativas e Motivações

Os principais fatores que influenciaram na escolha do tema da pesquisa estão relacionados com a mobilidade e, principalmente, com a questão da segurança encontrada nas redes *Wireless*. Por se tratar de uma tecnologia relativamente recente, muitas de suas vulnerabilidades já foram encontradas e muitas outras ainda serão descobertas.

As redes *Wireless* tornaram-se um alvo fácil e almejado por ataques de pessoas mal intencionadas, que buscam o comprometimento de sistemas. Os protocolos de roteamento para esse tipo de tecnologia, em sua maioria, foram idealizados considerando ambientes seguros, levando mais em conta a facilidade em obter acesso à rede guiada através de uma conexão de rede *Wireless* e, principalmente, a falta de conhecimento técnico do gerente desta rede.

Fator determinante da segurança em redes *Wireless* está relacionado com a origem dos ataques, já que, em qualquer posição dentro da área de cobertura da rede em questão, pode originar-se um ataque, o que dificulta a tarefa de localização precisa da origem deste ataque. Por outro lado, ataques direcionados às redes *Wireless,* além de comprometerem os recursos destas, podem comprometer os recursos de outras redes com as quais estas forem interconectadas. Para que os ataques dirigidos às redes *Wireless* possam ser identificados e as contramedidas possam ser tomadas de forma eficaz, é necessária uma análise das vulnerabilidades inerentes às redes 802.11x. Para tanto, é preciso um estudo exaustivo sobre os protocolos que dão suporte às mesmas. Com isso, as falhas nestes protocolos são apontadas e as mudanças cabíveis podem ser sugeridas.

1.3. Contribuições do Trabalho

A principal característica de uma rede sem fio *Ad Hoc* é o fato de não possuir infra-estrutura. Os dispositivos móveis (também conhecidos como nodos ou nós), que compõem uma rede sem fio desse tipo, são capazes de se comunicar uns com os outros, funcionando como roteadores. Os problemas de segurança encontrados nessas redes estão diretamente relacionados à maneira como funcionam seus dispositivos móveis, tendo como principal característica confiar a cada um deles a tarefa de encaminhar os pacotes de informação, que trafegam pela rede, desde sua origem até o seu destino.

Sua topologia é dinâmica, sofrendo diversas mudanças devido à mobilidade de seus nós durante a existência da rede. O fato de ser formada por nós móveis é uma indicação de que, normalmente, esses equipamentos têm sérias restrições, quanto ao consumo de energia, para se manterem funcionando.

As questões de segurança vêm recebendo foco das pesquisas ano após ano e, dentro em breve, é muito provável que teremos nos aproximado da segurança oferecida pelas redes tradicionais. Diante deste cenário, é fundamental entender se tais propostas têm apresentado impacto sobre o funcionamento das redes *Ad Hoc*. O enfoque deste trabalho objetiva estabelecer uma referência entre os resultados obtidos por protocolos, ditos não seguros, com está nova visão de protocolos seguros.

1.4. Trabalhos Correlatos

A segurança de redes é uma questão amplamente difundida nos cenários de redes fixas. No entanto, em redes sem fio, esta temática ainda é um desafio, sobretudo em redes *Ad Hoc*.

Em alguns trabalhos, encontrados na literatura, os autores, focando um ou mais tipos de ataques, ao tempo em que realizaram a análise da segurança, empregaram técnicas e propuseram novos algoritmos de roteamento, a fim de resolver o problema de segurança.

O projeto de algoritmos de roteamento, aliás, é um problema bastante estudado em redes móveis *Ad Hoc*. Diversos algoritmos já foram propostos nas literaturas, tais como AODV

(*Ad Hoc On-Demand Distance Vector Routing*), DSR (*Dynamic Source, Routing*), LMR (*Lightweight Mobile Routing*), TORA (*Temporally Ordered Routing Algorithm*), ABR (*Associativity-Based Routing), SSR (Signal Stability Routing*), SEAD (*Secure Efficient Ad Hoc Distance Vector Routing Protocol*). De acordo com a MANET, as principais qualidades que um algoritmo de roteamento para redes *Ad Hoc* deve apresentar são: operar de forma distribuída; ser livre de loops de roteamento; ter operações baseadas em demanda de tráfego; ser seguro; possuir uma latência baixa; suportar links unidirecionais.

A maioria das pesquisas sobre algoritmos de roteamento para redes *Ad Hoc,* propostos até então, focaram em suprir as necessidades de comunicação, assumindo os ambientes como confiáveis. Entretanto, com a expansão das suas aplicações, surgiu a necessidade de reconsiderar os problemas de roteamento e comunicação segura.

Em Hu [Hu, 2002] é apresentado um protocolo reativo para roteamento de redes *Ad Hoc*, chamado Ariadne. Este protocolo foi idealizado levando em consideração os requisitos propostos pela MANET, dando grande ênfase aos quesitos de segurança.

Pelo mesmo autor [Hu, 2002b] é apresentado um protocolo pró-ativo chamado SEAD, considerado seguro para redes *Ad Hoc*, que faz uso do roteamento por vetor de distância. Este protocolo também foi idealizado levando em consideração os requisitos propostos pela MANET, dando ênfase principal aos quesitos de segurança.

O protocolo SAODV (*Secure Ad Hoc On-Demand Distance Vector*) foi proposto por Zapata [Zapata, 2001] como uma extensão do protoloco de roteamento AODV. O protocolo SAODV pode ser usado para proteger os mecanismo de descoberta de rotas, provendo características seguras, tais como integridade, autenticação e não-repúdio.

Por Hu [Hu, 2003], foi proposto um protocolo de roteamento para tratar de ataques do tipo wormhole, onde o atacante grava pacotes em um local na rede; em seguida transmite o pacote tunelando-o para outro local, que o retransmite para a rede. Este é um dos ataques mais severos e difíceis de serem identificados que ocorrem em uma rede *Ad Hoc*. A maioria dos protolocos sem mecanismos de proteção contra ataques wormhole seria incapaz de encontrar uma rota maior que um ou dois saltos sem severas interrupções de comunicação. O protocolo proposto é chamado de TIK.

1.5. Organização Do Trabalho

Para explicitar e dar direcionamento e fundamentação à pesquisa, no primeiro capítulo apresentou-se a parte introdutória, onde foram descritos os objetivos, as justificativas, as motivações, a contribuição esperada do trabalho, trabalhos correlatos e a organização do mesmo.

No capítulo segundo, serão abordadas as tecnologias envolvidas nas redes *MANET*, as características incorporadas no padrão IEEE 802.11.

No capítulo 3, serão abordados detalhes dos protocolos de roteamento, distribuídos em 3 grandes classes, pró-ativos, reativos ou sob-demanda e os protocolos híbridos. Nesta seção abordaremos as características dos principais protocolos de cada uma das 3 grandes classes.

Já no capítulo 4, serão explorados os tipos de ataques que assolam as redes *MANET*.

O ambiente de simulação, os tipos de simulação e a ferramenta de simulção serão descritos no capítulo 5.

No capítulo 6, apresentaremos as simulações e os resultados.

No sétimo e último capítulo, serão apresentandas as conclusões e perspectivas de trabalhos futuros.

2. *MANET*

O conceito de uma rede *Ad Hoc* data do início da década de 70, quando a U.S DARPA (*United States Defense Advanced Research Projects Agency*) iniciou o projeto PRNET (Packet Radio Network), para explorar o uso de redes de pacote de rádio num ambiente tático para comunicação de dados. Mais tarde, em 1983, a DARPA lançou o programa SURAN (*Survivable Adaptive Network*) para expandir a tecnologia desenvolvida no projeto PRNET para suportar grandes redes e para desenvolver protocolos de rede adaptativos, os quais pudessem adaptar-se às rápidas mudanças de condições em um ambiente tático. O último da série dos programas iniciados pela DARPA, para satisfazer os requisitos de defesa para sistemas de informações robustos e rapidamente expansíveis, foi o GloMo (Global Mobile Information Systems), que teve início em 1994. Embora as comunicações táticas militares permaneciam a principal aplicação das redes *Ad Hoc*, havia um número crescente de aplicações não militares, tais como conferência, busca e salvamento.

A rede móvel *Ad Hoc,* também conhecida como *MANET* (*Mobile Ad Hoc Networt),* é uma rede *peer-to-peer* (servidores não centralizados) estabelecida temporariamente para suportar algumas necessidades imediatas. Em uma rede *MANET*, todos os terminais movem-se livremente e não necessitam de nenhuma infra-estrutura de comunicação. Entretanto, um terminal móvel somente consegue transmitir informações para outro terminal que esteja dentro da sua área de transmissão (terminal vizinho). Conseqüentemente, um pacote que deva alcançar um nodo de destino, que está fora da área de transmissão do nodo de origem, deverá passar por outros nós, chamados de nós intermediários, que servirão como roteadores, até chegar ao seu destino. Por este fato, um dos maiores desafios em redes móvel *Ad Hoc* está relacionado com a transmissão eficiente de pacotes de dados de um terminal móvel para o outro terminal.

Cada nodo participante na rede móvel *MANET* utiliza um protocolo de roteamento que permite descobrir caminhos com múltiplos saltos (*multi-hop*) para qualquer outro nó. Essa rede pode também ser chamada de rede sem infra-estrutura, já que os nós estabelecem rotas dinâmicas entre eles para formarem sua própria rede. A responsabilidade por organizar e controlar a rede é distribuída entre os próprios terminais. Em redes *MANET*, alguns pares de terminais não são capazes de se comunicar diretamente entre si, sendo, então, necessária alguma forma de retransmissão de pacotes (figura 5), para que assim estes pacotes sejam entregues ao seu destino.

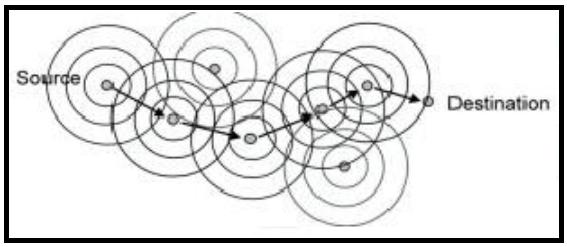


FIGURA 1 – RETRANSMISSÃO DE MENSAGENS

Exemplos de possíveis utilizações de uma rede móvel *MANET* podem incluir estudantes, utilizando laptops para participarem de uma conferência interativa, empresários, compartilhando informações durante uma reunião, comunicação em feiras e exposições e, soldados, enviando informações em campos de batalhas.

FIGURA 2 – MODELO DE COMUNICAÇÃO COM REDES MÓVEIS *MANET*.

Conforme apresentado na figura 2, os dispositivos computacionais são capazes de trocar informações diretamente entre si. As redes *MANET* são, principalmente, indicadas para situações onde não se pode, ou não faz sentido, instalar uma rede fixa. Neste tipo de rede, uma rota entre dois computadores pode ser formada por vários *hops* (saltos) através de um ou mais computadores na rede.

As redes *MANET* podem ainda ser subdivididas em outras duas categorias: redes de comunicação direta (os nós da rede somente se comunicam com outros nós que estejam dentro do seu raio de cobertura) e redes de múltiplos saltos (os nós móveis se comportam como roteadores, permitindo que se comuniquem, mesmo que a distância entre origem e destino seja maior do que o raio de cobertura). Redes *MANET* de múltiplos saltos são mais complexas do que as de comunicação direta.

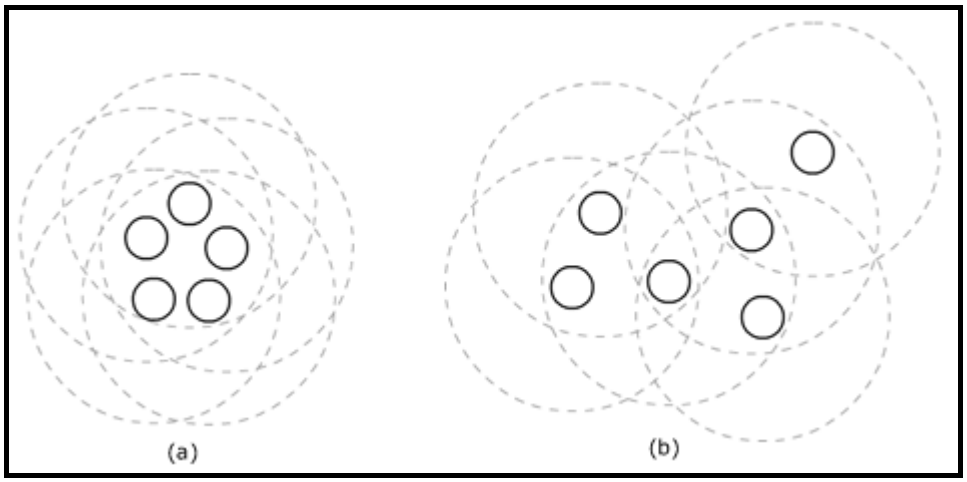


FIGURA 3 – DOIS TIPOS DE REDES *AD HOC*: (A) COMUNICAÇÃO DIRETA, (B) MÚLTIPLOS SALTOS.

As redes *MANET* têm vantagens e desvantagens, se comparadas com redes infra-estruturadas [Câmara, 1999], [Câmara, 2001].

Vantagens:

- **Rápida Instalação**: as redes podem ser instaladas rapidamente em locais sem nenhuma infra-estrutura prévia pelo fato de que não necessitam de bases fixas para rotear as mensagens;
- **Tolerância à falhas**: o mau funcionamento ou desligamento de uma estação pode ser facilmente resolvido com a reconfiguração dinâmica da rede. Em uma rede fixa, ao contrário, quando ocorre uma falha em um roteador, o redirecionamento de tráfego é uma operação complexa, quando possível, ou seja, quando há caminhos suficientes para isto. Falhas em redes infra-estruturadas são mais graves, principalmente se ocorrerem na Estação de suporte à mobilidade, pois trata-se de uma entidade centralizadora, isto é, todas as comunicações dos nós dependem dela;
- **Conectividade:** se duas estações estão dentro da área de alcance das ondas de rádio, elas têm um canal de comunicação. Em uma rede fixa, mesmo que duas estações estejam uma ao lado da outra, é necessário que as estações estejam ligadas por um meio guiado, para que troquem informações. Em uma rede infraestruturada, é necessária a comunicação do host com a estação base e, desta, para o outro host;
- **Mobilidade:** em contraposição à falta de mobilidade dos computadores fixos.

Desvantagens:

- **Banda passante:** canais de comunicação sem fio normalmente possuem bandas passantes menores e limitadas devido ao meio utilizado para a comunicação;
- **Erros no enlace sem fio:** a taxa de erros em um link sem fio é tipicamente de um bit errado a cada 105 ou 106 bits transmitidos, enquanto que em uma fibra óptica esta taxa é tipicamente de um a cada 1012 - 1015 bits retransmitidos;
- **Localização:** existe o problema de se conhecer a localização física do host móvel e enviar, para esse ponto, suas mensagens. Em redes fixas, este problema não ocorre, pois o endereço IP indica implicitamente a localização do nó. Já em redes *Ad Hoc*, localizar o usuário é um problema, pois não se tem nenhuma informação geográfica, sendo que o endereço da máquina não tem necessariamente mais nenhuma relação com sua posição;
- **Roteamento:** em uma rede fixa, a topologia dificilmente se altera. Em redes *Ad Hoc,* os nós movem-se de um lado para o outro de forma não determinística.

A IETF (Internet Engineering Task Force) tem um grupo de trabalho chamado *MANET* (Mobile *Ad Hoc* NETworks), que é responsável por discutir os problemas referentes às redes *Ad Hoc*. Foi publicada pelo grupo, em janeiro de 1999, a RFC 2501 [Corson, 1999], onde foram discutidas as principais características e problemas das redes *Ad Hoc*. De acordo com o grupo *MANET*, as características salientes que os algoritmos de roteamento para redes *Ad Hoc* devem apresentar são:

- **Topologia dinâmica:** os nós são livres para se moverem arbitrariamente, ou seja, a topologia da rede muda freqüentemente e os links podem ser bidirecionais ou unidirecionais;
- **Tamanho de banda restrito:** links sem fio continuam a ter capacidade significativamente menor do que links das redes fixas;
- **Operação para economia de energia:** alguns ou todos os nós numa MANET utilizam baterias para fornecimento de energia; neste sentido, um dos critérios mais importantes de desenvolvimento do sistema deve ser a conservação de energia;
- **Segurança física limitada:** as redes móveis sem fio são mais vulneráveis às ameaças à segurança do que as redes fixas; o aumento da possibilidade de escuta, invasão e ataques devem ser cuidadosamente considerados; em contrapartida, por terem um controle descentralizado, possuem como benefício terem maior robustez em relação às redes centralizadas.

Um dos principais problemas existentes em um ambiente de comunicações móveis está ligado à possibilidade de perda de comunicação entre dois nós, como conseqüência da degradação do nível do sinal recebido e da interferência presente no meio rádio. Dessa forma, a comunicação entre um nodo e seus vizinhos deve sempre poder ser restaurada, por exemplo, através de outros nós, segundo um mecanismo de roteamento. Outro problema surge a partir do momento em que se deseja enviar uma mensagem para um outro nó, pois é necessário encontrar o nodo de destino para então traçar uma rota até ele e manter esta rota até o fim da comunicação. Desta forma, se faz necessário um mecanismo de gerenciamento de localização.

Existem várias razões para que o roteamento e a localização de um móvel em redes móveis sem fio seja mais complexo do que em redes fixas:

- **Mobilidade e topologia de rede dinâmica -** Tanto as características do canal quanto a mobilidade dos nós podem alterar as conexões entre os nós e, assim, mudar a topologia da rede. Os protocolos de rede devem ser capazes de manter a operação normalizada enquanto a topologia da rede muda com o tempo;
- **Throughput** - Já que o espectro é um recurso escasso, throughput é definitivamente uma das considerações críticas no projeto dos protocolos de roteamento. Em se tratando de aplicações multimídia de tempo real, esta consideração fica mais crítica;
- **Retardo** - Características de retardo são importantes para todos os tipos de aplicações, mas, especialmente para aquelas limitadas no tempo e aplicações multimídia, tais como voz e vídeo;
- **Interferência com outros nós móveis** - Como os nós podem movimentar-se aleatoriamente e, conseqüentemente, podem aproximar-se muito uns dos outros, as interferências entre os nós aumentam, principalmente se houver reuso espacial.
- **Inexistência de uma entidade central** - A falta de uma entidade centralizadora, com a capacidade de coordenar a rede de forma completa, exige a adoção de sofisticados algoritmos de roteamento, o que de fato torna a forma de operação mais complexa.
- **Todas as comunicações devem ocorrer através do meio sem fio** - Este fator traz vários problemas relativos à conectividade, à propagação de sinais e à baixa velocidade do canal.
- **Consumo de energia -** O suprimento de energia do nodo móvel normalmente é dependente de baterias e estas têm um suprimento limitado de energia. Uma das características mais desejáveis de algoritmos de roteamento em redes móveis, ao lado do tempo de convergência, é a economia de energia.

3. Roteamento em Redes MANET

Em 1999, o grupo IETF - *Internet Engineering Task Force* introduziu algumas propriedades que devem ser levadas em conta quando tratamos da qualidade dos protocolos de roteamento em uma rede *Ad Hoc*, tais como:

Operação distribuída: propriedade essencial para o roteamento em uma rede *Ad Hoc* para evitar a centralização que leva à vulnerabilidade;

Protocolos livres de loops: para que os pacotes não fiquem trafegando durante um período de tempo relativamente grande na rede, pode ser usada como solução uma variável do tipo TTL (*time to live*), mas, uma abordagem melhor estruturada é mais indicada;

Operação baseada na demanda: o algoritmo de roteamento deve ser adaptável às condições de tráfego; se isto for feito de forma inteligente, os recursos de energia e largura de banda serão utilizados de forma mais eficiente;

Operação pró-ativa: em alguns momentos, a latência adicionada pela operação baseada na demanda poderá ser inaceitável; se os recursos de energia e largura de banda permitirem, operações pró-ativas serão desejáveis;

Segurança: se as camadas de rede e de enlace não garantirem segurança, os protocolos de roteamento MANET estarão vulneráveis a muitas formas de ataques; é necessário que haja mecanismos para inibir modificações nas operações dos protocolos;

Operação nos períodos de "sonolência" do nó: como resultado da conservação de energia ou de alguma outra inatividade, os nós devem parar de transmitir e/ou receber pacotes, por um período arbitrário de tempo, sem que isto resulte em maiores conseqüências;

Suporte a links unidirecionais: tipicamente, uma rede *MANET* assume links bidirecionais e muitos algoritmos são incapazes de funcionar corretamente sobre links unidirecionais. Entretanto, links unidirecionais podem ocorrer em redes sem fio.

Segundo o grupo *MANET*, existem pontos que podem ser observados para avaliar, de forma quantitativa, um protocolo de roteamento para redes *MANET* [MANET, 1999]. São eles:

I. *Throughput* e atraso fim-a-fim dos pacotes de dados;

II. Tempo para aquisição de uma rota, principalmente para algoritmos que trabalham sob demanda de rotas;
III. Porcentagens de pacotes entregues fora de ordem;
IV. Eficiência do protocolo, que é o número de pacotes de controle necessários para que o protocolo funcione corretamente;
V. Capacidade de manipular e escolher a melhor rota baseado em parâmetros QoS.

3.1. Algoritmos de Roteamento para uma Rede MANET

Nesta seção, abordaremos alguns algoritmos de roteamento para uma rede *Ad Hoc*. Os algoritmos podem ser classificados de várias maneiras, dependendo de suas características físicas e/ou lógicas. Apresentaremos uma taxonomia que pretende abranger duas características essenciais dos algoritmos, no que diz respeito à topologia lógica da rede e à construção das rotas. A partir disto poderemos considerar os algoritmos de roteamento para uma Rede *Ad Hoc* da seguinte maneira:

Filosofia do Roteamento:

Roteamento Flat: todos os nós são "iguais" e o roteamento dos pacotes é feito baseado em conexões ponto-a-ponto, restringidas somente pelas condições de propagação. A figura 4 ilustra este roteamento [Haas , 1997], [Haas , 1998].

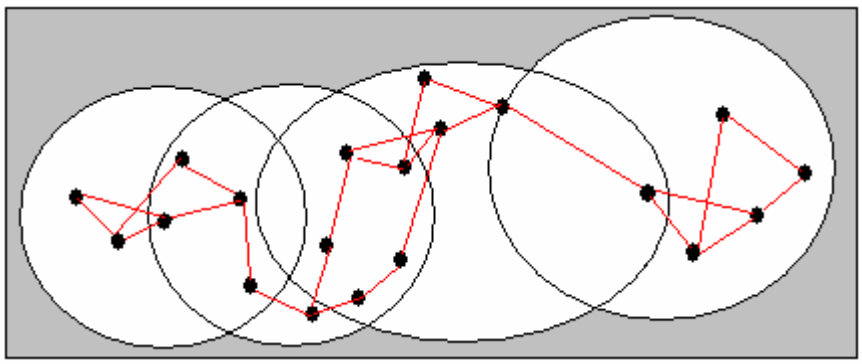


FIGURA 4 – ROTEAMENTO FLAT, [HAAS , 1998].

Roteamento hierárquico: em um roteamento hierárquico, existem ao menos duas camadas; uma camada mais baixa, onde nós próximos geograficamente criam redes ponto-a-

ponto. Em cada uma destas redes, ao menos um nodo é designado para servir como "*gateway*" para a camada mais alta. Estes nós "*gateway*" criam a camada mais alta da rede, a qual, usualmente, requer maior poder de transmissão/recepção. A figura 5 ilustra um roteamento hierárquico.

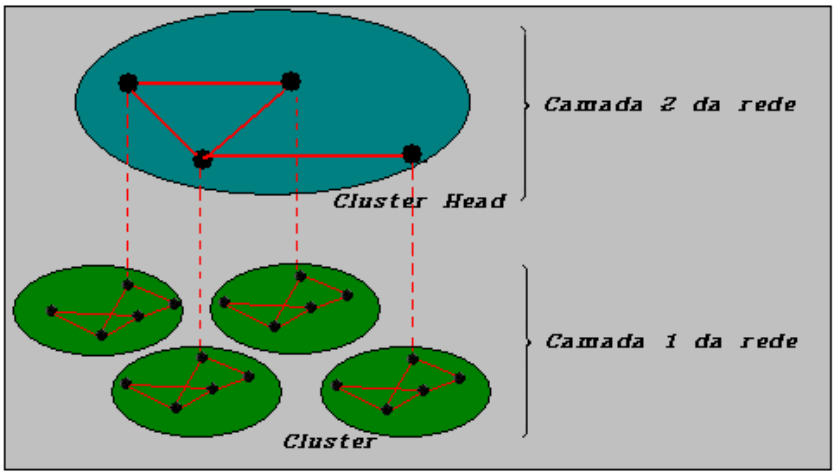


FIGURA 5 – ROTEAMENTO HIERÁRQUICO, [HAAS , 1998].

Construção das Rotas:

- **Reativos**, nos quais um nodo cria a rota somente quando requisitado, possibilitando economia de banda e de bateria e levando um tempo maior para a obtenção de rotas;
- **Pró-ativos**, nos quais um nodo atualiza, de forma periódica, as rotas;
- **Híbridos**, em que apenas uma parte dos nós faz atualização periódica.

3.2. Protocolos de roteamento pró-ativos

Nos protocolos de roteamento pró-ativo (table-driven), o algoritmo tenta avaliar continuamente as rotas de modo que, quando um pacote necessitar de encaminhamento, a rota já seja conhecida e possa ser utilizada imediatamente. De forma semelhante aos protocolos para redes fixas (cabeadas), cada nó, a todo o momento, possui informação em sua tabela referente a todos os possíveis destinos. As informações são mantidas mesmo que o nodo, onde o protocolo está sendo executado, nunca tenha utilizado muitas dessas rotas, tanto para enviar seus próprios

pacotes como para enviar pacotes de outros nós, quando faz o papel de roteador. Para isto, os nós mantêm uma ou mais tabelas com informações referentes à rede e respondem a mudanças na sua topologia, propagando atualizações de modo a manter a consistência da rede. Estas atualizações são iniciadas por mecanismos de temporização, o que faz com que haja sempre um número constante de transmissões em andamento, mesmo quando a rede esteja em equilíbrio [Zhou, 2004].

A diferença entre estes protocolos está na maneira como a informação de roteamento é atualizada e detectada e o tipo de informação mantido em cada tabela de roteamento. Além disso, cada protocolo de roteamento pode manter um número diferente de tabelas. Normalmente, esse tipo de protocolo consegue ter um melhor desempenho, sendo mais veloz, no tempo de resposta para o nodo origem, que solicitou uma determinada rota, do que os protocolos reativos. Dado que todas as rotas possíveis devem existir na tabela de roteamento de cada nó, nesta seção faremos a descrição das características importantes dos principais protolos pró-ativo.

3.2.1. *Secure Efficient Ad Hoc Distance Vector Routing Protocol* (SEAD)

O protocolo SEAD é um protolo de roteamento pró-ativo e seguro para redes *Ad Hoc* que faz uso do roteamento por vetor de distância. Muitos protocolos de roteamento para redes *Ad-Hoc* já foram implementados, baseados no vetor de distância, mas em geral os ambiante foram tratados como ambiente confiável, levando em consideração apenas questões de roteamento.

O projeto do SEAD [Hu, 2002] implementou primitivas criptográficas, em cada parte de suas funcionalidades, para criar um protocolo eficiente, prático, que fosse robusto contra múltiplos tipos de ataques sem coordenação, os quais podem gerar estados incorretos em qualquer nodo da rede. Juntamente com a segurança gerada pela aproximação existente entre a camada física e a camada MAC, na pilha de protocolo de rede, o protocolo provê uma base para a operação segura de uma rede *Ad-Hoc*.

A implementação do protocolo SEAD foi inspirada no DSDV-SQ, uma versão do protocolo DSDV. O DSDV-SQ apresenta um melhor desempenho, como ficou demonstrado em

comparações realizadas com outras versões do protocolo DSDV, através de simulações realizadas, considerando redes *Ad Hoc*.

Este protocolo trata de ataques que modificam informações de roteamento transmitidas por *broadcast* durante a fase de atualização do protocolo DSDV-SQ. Desta forma, o roteamento pode ser interrompido se o ataque modificar os campos *sequence number* ou *metric* da mensagem de atualização da tabela de roteamento. *Replay attacks* também são tratados pelo SEAD.

Para garantir segurança no protocolo DSDV-SQ, o protocolo SEAD faz uso de cadeias de hash unilaterais eficientes, ao invés de retransmitir, usando operações de criptografia assimétrica. Entretanto, o protocolo SEAD assume que algum nodo tenha um mecanismo para distribuir elementos autenticados da cadeia de hash, os quais podem ser usados para autenticar todos os outros elementos da cadeia. Uma solução tradicional é garantir que a distribuição de chaves seja realizada por uma entidade confiável, que assina certificados de chave pública para cada nodo; cada nodo, então, poderá usar sua chave pública para assinar um elemento da cadeia de hash e distribuí-lo [Hu, 2002] .

A idéia básica do protocolo SEAD consiste, pois, em autenticar o *sequence number* e o *metric* de uma mensagem de atualização da tabela de roteamento, usando elementos da cadeia de hash. Além disso, o receptor das informações de roteamento SEAD também autentica o remetente, assegurando que as informações de roteamento foram originadas do nodo correto.

Cada nodo usa um elemento autêntico, específico de sua cadeia de hash, em cada atualização de roteamento, a qual é enviada para o próprio nodo (*metric* 0). Baseado neste elemento inicial, a cadeia de hash unilateral provê autenticação para o salto mais baixo no campo *metric* em outras atualizações de roteamento para este nodo. O uso do valor de hash correspondente ao *sequence number* e ao *metric* em uma entrada de atualização de roteamento previne que qualquer nodo anuncie uma rota para algum destino, reivindicando um *sequence number* maior que o *sequence number* do próprio destino. Igualmente, um nodo não pode anunciar uma rota melhor que aquelas anunciadas, pois o *metric* existente em uma rota não pode ser diminuído devido à natureza da cadeia de hash unilateral [Hu, 2002].

Quando um nodo recebe uma atualização de roteamento, este confere a autenticidade da informação de cada entrada na atualização, usando o endereço do destinatário, o *sequence number* e o *metric* da entrada recebida juntamente com o último valor de hash recebido da cadeia

de hash do destinatário. Depois é feito o Hashing dos elementos recebidos quantas vezes for o correto, confirmando a autenticidade da informação recebida, caso o valor de *hash* calculado e os valores de *hash* autênticos forem iguais [Hu, 2002].

A fonte de cada mensagem de atualização de roteamento no SEAD também deve ser autenticada; caso contrário, pode permitir a criação de loops através do ataque de personificação (*impersonation attack*). Duas maneiras são propostas para realizar a autenticação: a primeira é baseada em um mecanismo de autenticação por broadcast, como o TESLA, e o segundo se baseia no uso de Códigos de Autenticação de Mensagem, assumindo a existência de uma chave secreta, compartilhada entre cada par de nodos na rede [Hu, 2002].

3.2.2. Destination-sequenced distance vector (DSDV)

O protocolo DSDV [Perkins, 2001], [Perkins, 2003] é uma modificação do DBF, que garante rotas livres do laço. Fornece um único trajeto a um destino, que seja selecionado usando o algoritmo mais curto do roteamento do trajeto do vetor da distância. Cada nodo da rede possui uma tabela com as informações que serão enviadas, por *broadcast*, para todos os demais nós da rede. Também possue uma tabela de roteamento com todas as rotas para cada um dos nós da rede e a quantidade de saltos para alcançar cada destino. Mesmo as rotas que não estão sendo utilizadas são mantidas na tabela. As entradas na tabela de roteamento são identificadas através de um campo chamado *sequence number*, sendo que o valor desse campo é informado pelo nodo destino, durante o processo de descoberta da rota. A manutenção da tabela é feita através do envio de mensagens periódicas por cada nó, informando as alterações que ocorreram em suas tabelas, devidas às mudanças na topologia da rede.

Existem dois diferentes tipos de mensagens para atualização das rotas:

Mensagens curtas, contendo apenas as últimas rotas que sofreram alguma modificação. Esse tipo de mensagem deve caber em um único NPDU (*Network Protocol Data Unit)*, diminuindo assim a quantidade de tráfego gerado por um *broadcast*.

Mensagens completas, contendo toda informação da tabela de roteamento. Mensagens desse tipo geram uma grande quantidade de tráfego. Para evitar uma sobrecarga da rede, essas mensagens devem ser enviadas com uma freqüência relativamente baixa.

Sempre que um nodo repassa a mensagem de *broadcast,* que recebeu sobre uma determinada rota, ele incrementa a quantidade de saltos e atualiza o número de seqüência, que é informado por cada nodo por onde a mensagem passa. Antes de retransmitir uma atualização de rota, o nodo aguarda um tempo, calculado com base na media do tempo que uma atualização mais recente pode variar antes de chegar. Esse retardo introduzido diminui a quantidade de tráfego gerado pelo nó.

Quando um nodo recebe informações sobre rotas que ele já possui, ele é capaz de diferenciar as novas informações das antigas, através do campo *sequence number*. A rota que possui o maior valor para o *sequence number* terá preferência. Aquela que possui o menor número será descartada ou mantida como uma rota alternativa de menor importância. Se os valores do campo *sequence number* forem iguais, é dada preferência para a rota com a menor quantidade de saltos para alcançar o destino.

Quando um determinado link se desfaz, o nodo que percebe essa mudança altera a entrada na sua tabela para essa rota, indicando uma quantidade de saltos igual ao maior valor possível para esse campo. Alterando também o valor do *destination sequence number,* esse é o único caso onde a alteração desse campo é feita por um nodo que não é o próprio destino. Por sua importância, essa alteração é imediatamente propagada pela rede pelo nodo que primeiro a percebeu, ou seja, essa mensagem de atualização não aguarda o momento em que as rotas modificadas são periodicamente disseminadas. Do mesmo jeito é tratada a atualização de uma rota que estava inacessível, sendo agora possível acessá-la.

A figura 6 ilustra a tabela de roteamento de um nodo e a posição desse mesmo nodo numa rede *Ad Hoc*:

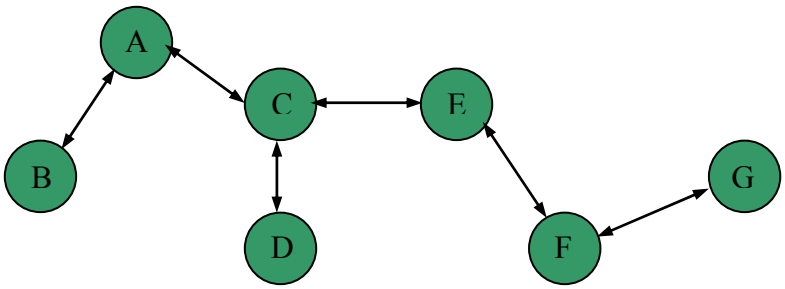


FIGURA 6 – REDE *AD HOC* UTILIZANDO PROTOCOLO DSDV, [PERKINS, 2003]

Destino	Próximo nó	Qtde Saltos	Nº de Seqüência	Criação
A	A	1	12	1
B	A	2	14	1
C	C	0	0	1
D	D	1	10	2
E	E	1	12	2
F	E	2	18	1
G	E	3	14	2

TABELA 2: TABELA DE ROTEAMENTO DO NODO C

Observação: Os valores para os campos, números de seqüência e criação, foram escolhidos ao acaso, servindo apenas a título de exemplo.

3.2.3. Wireless routing protocol (WRP)

O protocolo WRP [Murthy, 1996] é um protocolo pró-ativo que garante evitar os loops e os loops temporários em roteamentos através do uso de informação do predecessor. Este protocolo também faz com que cada nodo mantenha quatro tabelas:

- Tabela de Distância
- Tabela de Roteamento
- Tabela de Custo de Enlace
- Tabela de Lista de Retransmissão de Mensagem (MRL).

Os nós se informam das mudanças de enlaces através de mensagens de atualização. Estas mensagens são trocadas somente entre vizinhos e contém as informações das atualizações, além de uma lista indicando quais nós devem reconhecer as atualizações. A MRL guarda as atualizações contidas em uma mensagem de atualização, que precisam ser retransmitidas, e quais nodos vizinhos devem reconhecer a retransmissão. Se ocorrer a queda de um enlace entre dois nós, eles a notificam a seus vizinhos, que modificam suas tabelas de distância e procuram novos caminhos para outros nós. Quaisquer novos caminhos descobertos serão enviados de volta para os nós originais para que eles atualizem suas tabelas. Os nós aprendem sobre a existência de seus vizinhos através de reconhecimentos e outras mensagens. Se um nodo não estiver enviando mensagens, ele deve enviar periodicamente um *hello* para assegurar a conectividade. Este comportamento introduz uma quantia significativa de *overhead* na memória a cada nó, enquanto o tamanho da rede aumenta. Isso também consumirá uma quantia significativa de largura de

banda e de energia, já que cada nodo deverá permanecer ativo todas as vezes (ou seja, não poderá assumir o modo *sleep* ou espera, para conservar sua energia).

3.2.4. Global state routing (GSR)

O GSR [Cheng, 1998] é um algoritmo de roteamento baseado em *Link State* [Tanenbaum, 1996], isto é, o nodo de origem envia informações de sua tabela de roteamento para todos os nós, necessitando, para isso, de mecanismos sofisticados e eficientes para controlar o tráfego gerado. Assim, o GSR consegue manter uma visão global da rede, podendo fazer cálculos precisos de rotas. Algumas características importantes, apresentadas por este protocolo, são: utilização de técnicas para diminuir o tamanho das tabelas de roteamento e a sua capacidade de manipular parâmetros de *QoS*, o que muitos outros algoritmos de roteamento ainda não exploraram.

No GSR, quando um nodo detecta uma mudança de topologia, causada pelo mau funcionamento ou o desligamento de um nó, esta informação é mantida em uma tabela de topologia, onde já devem estar armazenadas informações de alterações recebidas pelos seus vizinhos. Estas informações são preparadas e disseminadas periodicamente apenas para os seus vizinhos e, com isto, ao contrário do *Link State*, não realiza a inundação (*flooding*) das informações de roteamento. Além desta tabela, cada nodo possui também uma lista de vizinhos, uma tabela de próximo nodo, e uma tabela de distâncias, e, como no estado inicial a lista de vizinhos e a tabela de topologia ficam vazias, será necessário atribuir valores de inicialização para as tabelas internas, que, com isto, poderão ouvir o meio para reconhecer seus vizinhos.

Para realizar redução do tamanho das mensagens de atualização, o GSR utiliza duas técnicas que são: *Fresh Update* e *Fisheye*.

Na *Fresh Update,* somente as informações consideradas úteis aos vizinhos serão disseminadas, isto é, se o seu vizinho contém uma informação mais recente, não faz sentido enviar sua informação para ele. Para a realização desta técnica, é necessário que cada nodo contenha um Vetor de Seqüência de Números (VSN) que, além de armazenar todos os destinos que estão na tabela, indica a idade de cada informação enviada. Com isto, o nodo destino

compara seu VSN com os valores recebidos dos vizinhos e, se a entrada for mais antiga, a mensagem de atualização é removida, ou seja, existe um *timeout* associado à informação.

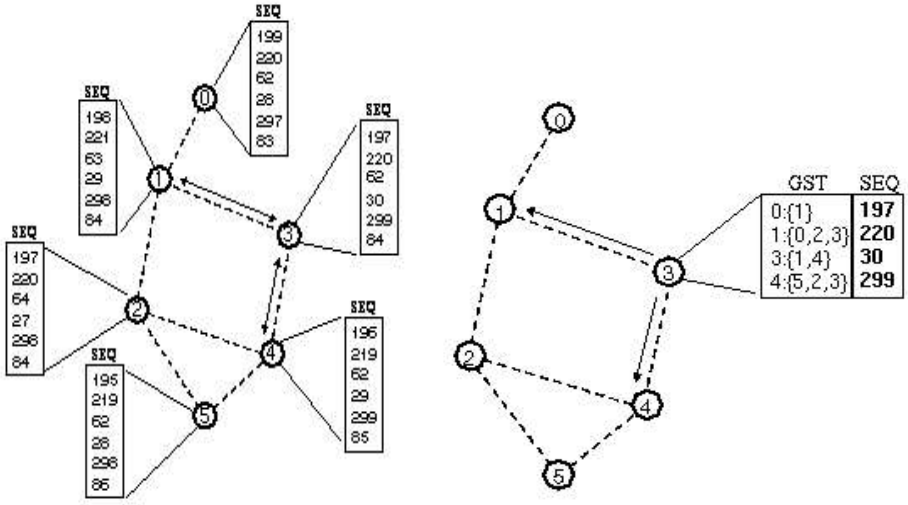


FIGURA 7 – RESULTADO DO MÉTODO FRESH UPDATE, [CHENG, 1998]

A outra técnica conhecida é o *Fisheye,* que tem por objetivo manter um volume alto de informações sobre os nós mais próximos e diminuir o detalhe das informações dos nós mais distantes.

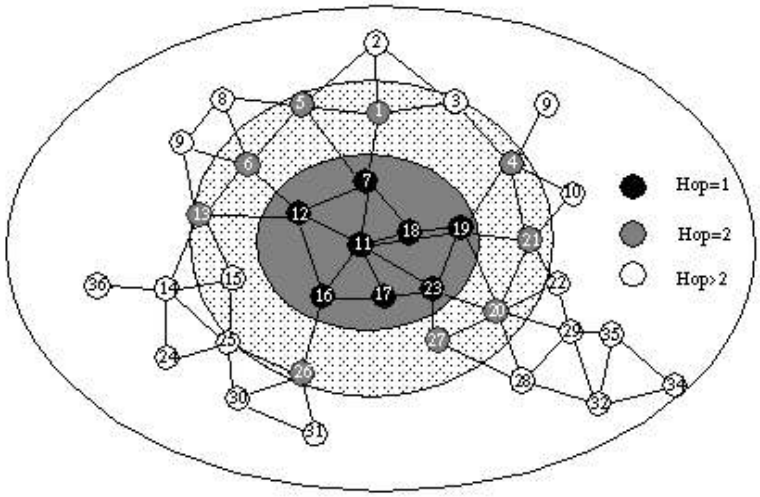


FIGURA 8 – MÉTODO FISHEYE COM DISTÂNCIA DE 1 NODO, [CHENG, 1998]

3.2.5. Fisheye state routing (FSR)

O protocolo FSR [Gerla, 2002] é descendente de GSR e, como ele, também está basedo no protocolo de rotemento *lisk state* utilizado nas redes fixas tradicionais. Trata-se de um protocolo pró-ativo ou table-driven. O FSR introduz a noção de escopo multi-nível para reduzir o *overhead* de atualização de rotas em grandes redes, o que o torna mais escalável. Entretanto, a escalabilidade vem ao preço de precisão reduzida. Isto acontece porque, com aumentos de mobilidade, as rotas para destino remoto ficam menos precisas. Isto pode ser superado fazendo a freqüência, através da qual são enviadas atualizações a destinos remotos, proporcional ao nível de mobilidade. Um nodo armazena o estado de Enlace (*Link State*) para muitos destinos na rede. Periodicamente, são enviados *broadcasts* de atualização de "Estado de Enlace" de um destino a seus vizinhos, com uma freqüência, que depende da distância, em saltos para esse destino (isto é, o "espaço" relativo a esse destino). São propagadas atualizações de estado que correspondem a destinos mais distantes, com freqüência mais baixa que para os destinos próximos. Através das atualizações de estado, os nós constroem o mapa da topologia da rede inteira e computam rotas eficientes. Como conseqüência, a rota em que o pacote viaja torna-se progressivamente mais exata na medida em que o pacote se aproxima do seu destino. As atualizações são propagadas como agregados, periódicamente (com o dependente do período na distância), em vez de cada fonte ser inundada individualmente.

3.2.6. Source-tree adaptive routing (STAR)

O protocolo STAR [Garcia, 1999] também é baseado no algoritmo *link state*. Cada roteador mantém uma árvore-fonte, que é um conjunto de *links* contendo os caminhos preferidos para os destinos. Esse protocolo diminui significativamente a quantia de *overhead* de roteamentos disseminada na rede, através do uso de uma *least overhead routing approach* (LORA – abordagem de roteamento *overhead* mínimo), para troca de informação de roteamento. Ele também sustenta uma *optimum routing approach* (ORA – abordagem de roteamento ótimo) para a hipótese de ser necessário. Essa abordagem elimina o processo periódico de atualização presente no algoritmo *Link State,* tornando a disseminação de atualização condicional. Como resultado, atualizações *Link State* são trocadas apenas quando certos eventos ocorrem. Portanto,

o STAR irá escalar bem em grandes redes, desde que tenha reduzido significativamente o consumo da largura de banda pelas atualizações de roteamento e, ao mesmo tempo a latência, através do uso de rotas pré-determinadas. Porém, esse protocolo pode precisar possuir memória significativa para o processamento de *overheads* em grandes redes e altamente móveis, porque cada nodo deverá manter um grafo da topologia parcial da rede (determinado a partir da árvore-fonte reportada por seus vizinhos), o qual pode variar freqüentemente enquanto os vizinhos continuam relatando informações de diferentes árvores-fonte.

3.2.7. Distance routing effect algorithm for mobility (DREAM)

Em comparação com os protocolos descritos até agora, o protocolo de roteamento DREAM [Basagni, 1998] emprega uma abordagem diferente para o roteamento. No DREAM, cada nodo reconhece suas coordenadas geográficas através de um GPS. Essas coordenadas são periodicamente trocadas entre cada nodo e, armazenadas em uma tabela de roteamento (chamada tabela de localização). A vantagem dessa troca de informações de localização decorre do fato de ela consumir muito menor largura de banda do que as trocas de informação *link state* ou *distance vector* completas, o que significa uma maior escalabilidade. No DREAM, o overhead no roteamento é posteriormente reduzido, quando a freqüência em que as mensagens de atualização são disseminadas, tornada proporcional à mobilidade e ao efeito da distância. Isso significa que os nós estacionários não precisam enviar nenhuma mensagem de atualização.

3.2.8. Multimedia support in mobile Wireless networks (MMWN)

No protocolo de roteamento MMWN, a rede é mantida com a utilização de uma hierarquia de *clusters*. Cada *cluster* tem dois tipos de nós móveis: *switches* e *endpoints*. Cada *cluster* também tem um *location manager* (LM – gerenciador de localização), que executa o gerenciamento de localização para cada *cluster* (ver figura 8). Toda a informação no MMWN é armazenada em uma base de dados dinamicamente distribuída. A vantagem do MMWN é que apenas os LMs executam a atualização e descoberta de localização, o que significa que o *overhead* no roteamento é significativamente reduzido, quando comparado aos tradicionais

algoritmos pró-ativos (*table-driven*) (tais como o DSDV e WRP). Entretanto, o gerenciamento de localização está intimamente relacionado à estrutura hierárquica da rede, tornando a descoberta e atualização da localização muito complexa. Isso porque, no processo de descoberta e atualização da localização, as mensagens precisam viajar através da árvore hierárquica dos LMs (*location managers*). As alterações na associação da hierarquia de *clusters* dos LMs também vão afetar a árvore de gerenciamento hierárquico e, introduzir um gerenciamento de consistência complexa. Essa característica apresenta problemas de implementação que são de difícil resolução. [Katz, 1994].

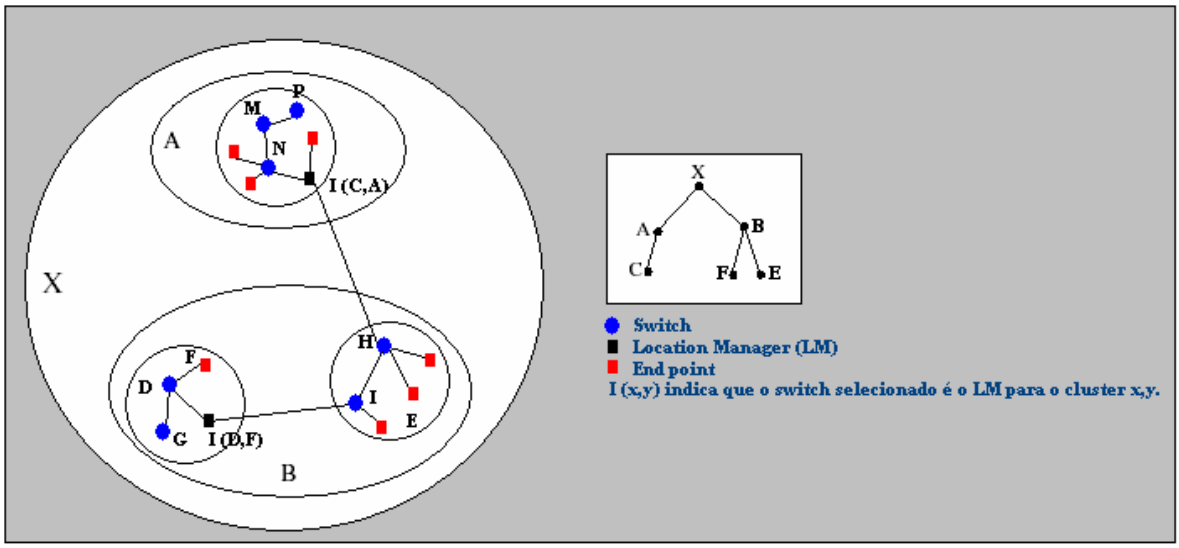


FIGURA 9 – UM EXEMPLOE OF CLUSTERING HIERARCHY IN MMWN, [KATZ, 1994].

3.2.9. Cluster-head gateway switch routing (CGSR)

O CGSR [Chiang, 1997] é outro protocolo de roteamento hierárquico onde os nós são agrupados em *clusters*. Entretanto, o esquema de endereçamento usado aqui é mais simples do que no MMWN. No CGSR, não há necessidade de manter uma hierarquia de *cluster* (que é necessária em MMWN). Ao contrário, cada *cluster* é mantido com uma *cluster-head*, que é um nodo móvel, eleito para gerenciar todos os outros nós dentro do *cluster* (Fig. 2). Esse nodo controla o meio de transmissão, sendo que todas as comunicações *inter-cluster* ocorrem através desse nó. A vantagem desse protocolo é que cada nodo mantém rotas apenas para seu *cluster-head*, ou seja, os *overheads* de roteamentos são mais baixos, se comparados à informação de roteamento por *flooding,* através de toda a rede. Entretanto, existem significativos *overheads* associados com a manutenção de *clusters*. Isso porque cada nodo precisa transmitir,

periodicamente, sua tabela de membro do *cluster* e, atualizar sua tabela com base nas atualizações recebidas.

Este protocolo, ao invés de utilizar uma rede organizada de forma plana, utiliza uma rede de nós agrupados, com vários esquemas heurísticos de roteamento. Utiliza-se um algoritmo distribuído dentro do grupo para eleger um nó *clusterhead*. Mudanças freqüentes do nó *clusterhead* podem comprometer seriamente a performance do protocolo, o que faz com que o algoritmo de agrupamento *Least Cluster Change* (LCC) seja utilizado. Este algoritmo determina que só sejam mudados os *clusterheads* quando dois deles se encontram ou, quando um nó perde contato com todos os clusterheads. O CGSR utiliza o DSDV como esquema de roteamento base, acrescido de um esquema hierárquico de roteamento *clusterhead-to-gateway*, no qual todo o tráfego para fora do grupo é roteado até o *clusterhead* e, dele, para outro *clusterhead*, através de um *gateway*. Cada nó possui duas tabelas: a Tabela de Membros de Grupo e a Tabela de Roteamento. Consultando as duas tabelas, o nó escolhe qual o *clusterhead* mais próximo e, através da Tabela de Roteamento, identifica como enviar o pacote até o *clusterhead*.

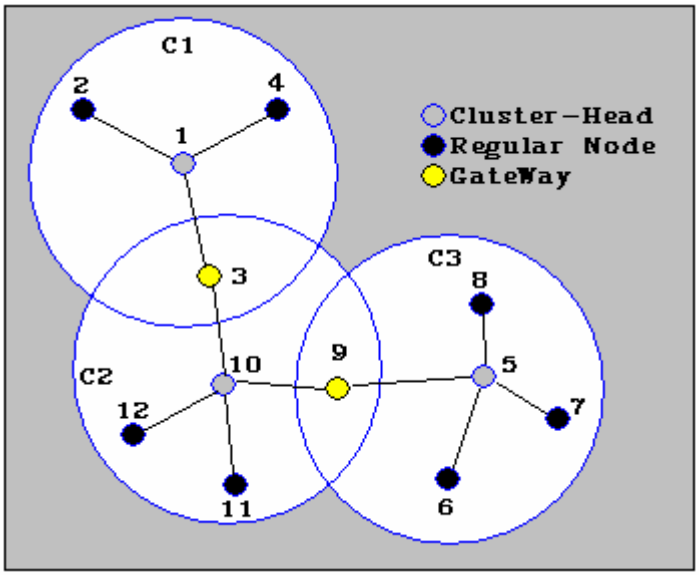


FIGURA 10 – ILUSTRAÇÃO DE UMA REDE BASEADA EM CLUSTER, [CHIANG, 1997]

3.2.10. Hierarchical state routing (HSR)

O HSR [Pei, 1999] é também baseado no tradicional algoritmo *Link State*. Entretanto, ao contrário dos outros algoritmos, baseados em *Link State,* descritos até agora, o HSR mantém

um endereçamento hierárquico e mapa topológico. O algoritmo de clusterização bem como o CGSR podem ser utilizados para organizar os nós com estreita proximidade nos *clusters*. Cada *cluster* possui três tipos de nós: o nodo *cluster-head*, que age como um coordenador de local para cada nó; nós *gateway*, que são nós que permanecem em dois *clusters* diferentes; e os nós internos, que são todos os outros nós em cada *cluster*. Todos os nós têm um único ID, que é normalmente o endereço MAC para cada nó. Os nós dentro de cada *cluster* transmitem suas informações de link uns aos outros. No HSR, cada nodo também tem um ID hierárquico (HID), que é uma seqüência dos endereços MAC da hierarquia topo para o nodo fonte. Por exemplo (veja figura 14), o HID do nodo 8 é h2; 2; 8i. O HID pode ser utilizado para enviar o pacote a partir de qualquer fonte para qualquer destino na rede. Assim, considerando o envio de um pacote do nodo 8 para o nodo 3: O nodo 8 tinha um HID de h2; 2; 8i e o nodo 3 tem HID de h4; 4; 3i. O pacote é, primeiramente, enviado para o nodo 2 (topo da hierarquia). O nodo 2, então, envia o pacote para o nodo 4, que é o topo de hierarquia do nodo 3. Os nós 2 e 4 formam um ‘‘link virtual’’, que é o caminho h2; 9; 5; 6; 4i. O nodo 4, por sua vez, enviará o pacote para o nodo 3. A clusterização lógica possibilita um relacionamento lógico entre clusterhead, em um alto nível. Aqui, os nós recebem endereços lógicos na forma < subrede; host >. Por exemplo, o nodo lógico 2 no nível 2, da figura 14, tem um endereço lógico h2; 2i. Os nós lógicos são conectados via links lógicos, os quais formam um “túnel” entre clusters de nível mais baixo. Nós lógicos trocam informação lógica de link bem como um resumo da informação dos clusters de nível mais baixo. Uma informação lógica de *Link State* é, então, enviada por *flooding* para os níveis mais baixos. Os nós físicos no nível mais baixo terão, assim, uma topologia ‘‘hierárquica’’ da rede. A vantagem do HSR sobre outros protocolos de roteamento hierárquico (como o MMWN) é a separação do gerenciamento de mobilidade da hierarquia física. Isso acontece via *Home Agents* (HA). Esse protocolo também tem muito menos controle de *overhead,* se comparado ao GSR e FSR. Entretanto, esse protocolo (de forma similar a qualquer outro protocolo baseado em cluster) introduz *overheads* extras na rede, a partir da formação e manutenção de clusters.

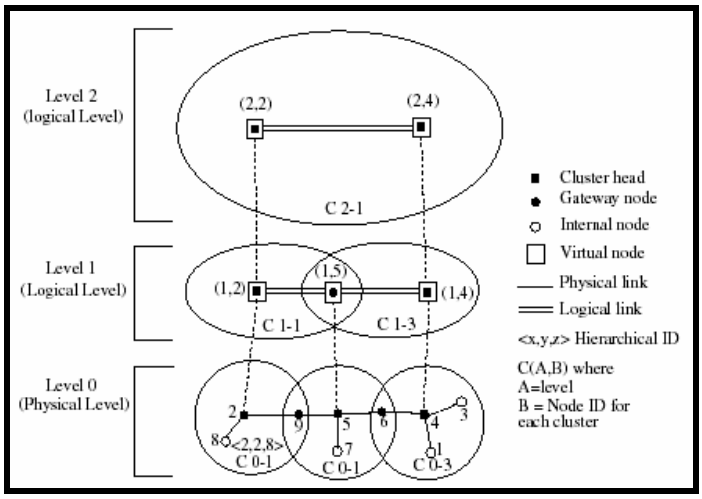


FIGURA 11 – EXEMPLO DE TOPOLOGIA HIERÁRQUICA, [PEI, 1999].

3.2.11. Optimised link state routing (OLSR)

O OLSR [Jacquet and Mühlenthaler, 2001], [Jacquet and Mühlenthaler, 2002] é um protocolo de roteamento ponto-a-ponto, baseado no tradicional algoritmo *Link-State,* pró-ativo. Nessa estratégia, cada nodo mantém informação de topologia sobre a rede através de trocas periódicas de mensagens de *Link-State*. A inovação do OLSR é que ele minimiza o tamanho de cada mensagem de controle e o número de nós retransmitindo, durante cada atualização de rota, através da utilização de uma estratégia de *multipoint replaying* (MPR). Para isso, durante cada atualização de topologia, cada nodo na rede seleciona um conjunto de nós vizinhos para retransmitir seus pacotes. Esse conjunto de nós é chamado de *multipoint relays* daquele nó. Qualquer nodo que não está no conjunto lê e processa cada pacote, mas não o retransmite. Para selecionar os MPRs, cada nodo transmite periodicamente uma lista dos seus vizinhos a um salto de distância utilizando mensagens *hello*. A partir da lista de nós nas mensagens *hello*, cada nodo seleciona um subconjunto vizinho a um salto de distância, o que cobre todos os vizinhos a dois *hops* (saltos) de distância. Por exemplo, na figura 12, o nodo A pode selecionar os nós B, C, K e N para serem os nós MPR,uma vez que esses nodos cobrem todos os nós que estão a dois saltos de distância. Cada nodo determina uma rota ótima (em termos de saltos) para cada destino conhecido, utilizando sua informação de topologia (a partir das tabelas de topologia e de vizinhança), armazenando essa informação em uma tabela de roteamento. Portanto, rotas para todos os destinos são imediatamente disponibilizadas quando a transmissão de dados começa.

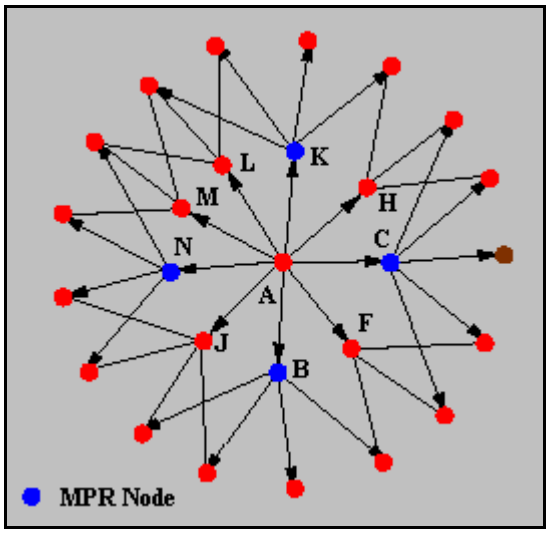


FIGURA 12 – MULTIPOINT RELAYS, [JACQUET AND MÜHLENTHALER, 2001].

3.2.12. Topology broadcast reverse path forwarding (TBRPF)

O TBRPF [Bhargav, 2001] é outro protocolo de roteamento, baseado em link-state, que executa roteamento *hop-by-hop*. O protocolo usa o conceito de *reverse-path forwarding* (RPF) para disseminar seus pacotes de atualização na direção reversa ao longo da árvore spanning, que é constituída pelo *minimum-hop path* (caminho com o mínimo de saltos) a partir dos nós em direção à fonte da mensagem de atualização. Nessa estratégia de roteamento, cada nodo calcula uma árvore fonte, que fornece um caminho para todos os destinos alcançáveis. Isso é feito através da aplicação de uma versão modificada do algoritmo Dijkstra's sobre a informação parcial de topologia, guardada em sua tabela de topologia. No TBRPF, cada nodo minimiza o *overhead,* relatando apenas uma parte de sua árvore fonte para seus vizinhos. A parte relatada de cada árvore fonte é trocada com os nodos vizinhos através de mensagens *hello* periódicas e diferenciadas. As mensagens *hello* diferenciadas relatam apenas as modificações do *status* dos nós vizinhos. Como resultado, as mensagens *hello* no TBRPF são menores do que nos protocolos que relatam a informação de link-state completa.

3.2.13. Resumo do roteamento pró-ativo

Em resumo, a maioria dos protocolos de roteamento pró-ativos de estrutura plana não se comporta muito bem, gerando o crescimento de escala da rede. Isso se deve ao fato de que os

procedimentos de atualização destes protocolos consomem uma quantia significativa da largura de banda da rede. O protocolo com estas características, que melhor se comporta diante desta suituação, é o OLSR. Para aumentar sua escalabilidade, trabalha-se pela redução do número de nós retransmitindo por *broadcast*, através do uso de *multipoint relaying*, que elege apenas um número de nós vizinhos para retransmitir a mensagem por *broadcast*. Abaixo, apresentamos uma tabela comparativa entre os protocolos.

Protocolo	Estrutura de Rotemento	Tabelas	Listas	Freqüência de atualização	Mensagens de Link-State	Nodo Crítico	Tempo de Convergência	Overhead de Memória
DSDV	Flat	2		Periódica e Requerida	Sim	Não	*O (D . I)*	*O(N)*
WRP	Flat	4		Periódica	Sim	Não	*O(h)*	*O(N^2)*
GSR	Flat	3	1	Periódica e Local	Não	Não	*O (D . I)*	*O(N^2)*
FSR	Flat	3	1	Periódica e Local	Não	Não	*O (D . I)*	*O(N^2)*
STAR	Hierárquico	1	5	Condicional – A atualização depende da ocorrência de algun evento	Não	Não	*O (D)*	*O(N^2)*
DREAM	Flat	1		Baseado na Mobilidade	Não	Não	*O (N . I)*	*O(N)*
MMWN	Hierárquico	Banco		Condicional – A atualização depende da ocorrência de algun evento	Não	Sim, gerenciamento de localização	*O (2D)*	*O(N)*
CGSR	Hierárquico	2		Periódica	Não	Sim, Cluster Head	*O (D)*	*O(2N)*
HSR	Hierárquico	2		Periódica dentro da subrede	Não	Sim, Cluster Head	*O (D)*	*O(N^2.L)+ O(S)+O(N/S)+O(N/n)*
OLSR	Flat	3		Periódica	Sim	Não	*O (D.I)*	*O(N^2)*
TBRPF	Flat	1	4	Periódica e Diferenciada	Sim	Sim, Nodo Pai	*O (D) ou D + 2 para Falhas de Link*	*O(N^2)+0(N)+O(N/V)*

TABELA 3 – COMPARAÇÃO DAS CARACTERÍSTICAS DOS PROTOCOLOS DE ROTEAMENTO PRÓ-ATVOS.

Fonte: Parte das características a serem comparadas foram extraídas de [Royer and Toh, 1999].

Protocolo	Controle de Overhead	Caracteristicas	Vantagens/Desvantagesn
DSDV	*O(N)*	Livre de Loop	Livre de Loop/Overhead alto
WRP	*O(N)*	Livre de loop, usando informação do predecessor.	Livre de loop/Overhead de memória
GSR	*O(N)*	Atualiza informações de localização. Mantem uma lista de todos os vizinhos disponíveis. Troca periódica de mensagens de link state com nodos vizinhos	Atualiza informações de localização/ Overhead de memória alto
FSR	*O(N)*	Controla a Frequência de atualização. Troca periódica de mensagens de link state com nodos vizinhos	Controla a Frequência de atualização/ Overhead de memória alto, precisão reduzida

STAR	*O(N)*	Minimiza o controle de overhead e através do uso do Lora e/ou Ora.	Minimiza o controle de overhead/Overhead de memória alto
DREAM	*O(N)*	Controla a taxa de atualização através da mobilidade e da distância.	Minimiza o controle de overhead e Overhead memória/Requer um GPS
MMWN	*O(X + E)*	Minimiza o controle de overhead e através do uso do Lora.	Minimiza o controle de overheade/Gerenciamento de movilidade e manutenção de cluster.
CGSR	*O(N)*	Troca de informação de roteamento baseada em cluster (clusterhead).	Controle de overhead reduzido/Formação e manutenção de cluster
HSR	*O(n.L)/I+ O(l)/J*	Baixo controle de overhead.	Minimiza o controle de overheade/Gerenciamento de localização.
OLSR	$O(N^2)$	Controle de overhead reduzido, baseado em multipoint relays	Conteção e controle de overhead reduzido/Requer o conhecimento de dois saltos vizinhos.
TBRPF	$O(N^2)$	Atualizações de topologia radiodifundindo em cima de uma árvore amplitude.	Controle de overhead pequeno/Overhead de memória alto.

(*l*)= um número fixo de tabelas atualizadas é transmitida V = número de notos vizinhos N= número de notos na rede N=número médio de nodos lógicos no cluster I – Intervalo médio de atualização D-Diametro da rede	S-númer de IP vitualr na subrede h-altura da arvore de roteamento X-número total number of LMs (cada cluster tem um LM); J-intervalo de registro de agente local para nodo L-número do nível hierarquico

TABELA 4 – COMPARAÇÃO DAS CARACTERÍSTICAS DOS PROTOCOLOS DE ROTEAMENTO PRÓ-ATVOS.

Fonte: Parte das características a serem comparadas foram extraídas de [Royer and Toh, 1999].

3.3. Protocolos de roteamento reativos (sob-demanda)

Protocolos de roteamento reativos são os protocolos que realizam o estabelecimento de uma rota apenas quando ela é solicitada pelo nó origem, ou seja, sob-demanda. Um processo de descoberta de rota é iniciado quando um determinado destino deve ser alcançado e não existe rota estabelecida para o mesmo. Esse processo é finalizado quando o destino é, finalmente, alcançado ou quando, após tentar todas as combinações de rotas possíveis, nenhuma é encontrada.

As rotas são mantidas na tabela de roteamento até que elas deixem de existir ou, após se passar um determinado tempo, sem que sejam utilizadas. Como o processo de descoberta de rota é realizado apenas quando uma origem qualquer solicita, os protocolos de roteamento reativos normalmente geram menos *overhead* que os pró-ativos, em detrimento do tempo que um nó deve esperar para ter a rota que solicitou estabelecida. Estes protocolos possibilitam economia de banda e de bateria, levando, porém um tempo maior para a obtenção de rotas. Se comparados com os protocolos pro-ativos, podemos dizer que esta forma de trabalhado dos protocolos reativos (sob-demanda) pode ser perigosa [Zhou, 2004]

3.3.1. Dynamic source routing (DSR)

O DSR [Johnson and Maltz, 1996] busca uma solução ao mesmo tempo simples e eficiente para redes *Ad Hoc*, que, fazendo uso de uma sobrecarga de roteamento muito baixa, permita que rotas sejam descobertas com reações rápidas a possíveis mudanças na topologia da rede. Assim, além de utilizar o roteamento sob demanda ou reativo, o protocolo usa o roteamento por fonte (nodo de origem), isto é, a fonte coloca, em cada pacote que envia, o caminho completo e ordenado que o pacote deve percorrer até se chegar ao destino. Esta aposta se justifica: o roteamento por fonte dispensa a necessidade de informações atualizadas de roteamento nos nós intermediários através dos quais o pacote passa em seu caminho até o destino e também constitui uma maneira trivial de se evitar loops.

Quando um nodo fonte deseja enviar pacotes a um nodo destino que não seja previamente conhecido, antes de mais nada precisa ser descoberta uma rota e, depois, enquanto se usa uma dada rota para o envio de pacotes, é necessário verificar se esta rota permanece válida. Caso contrário, o nodo fonte deve usar uma outra rota já conhecida, ou começar uma nova descoberta de rotas. Assim, caracterizam-se os dois mecanismos básicos do DSR: a descoberta de rotas e a manutenção de rotas. No DSR estes dois mecanismos agem completamente sob demanda, objetivando redução da sobrecarga de roteamento: não há necessidade de que qualquer tipo de pacote de qualquer camada seja trocado periodicamente na rede. Como conseqüência, o DSR não usa a distribuição periódica de informações para a atualização de tabelas de roteamento, como fazem os protocolos de roteamento pró-ativos, nem implementa qualquer mecanismo de mensagens periódicas para verificar a validade de um enlace, como faz, por exemplo, a implementação do AODV com o uso de mensagens de *hello*. Portanto, no DSR, a *overhead* de roteamento pode mesmo chegar a zero, quando nenhum nodo deseja transmitir pacotes ou quando os nós permanecem aproximadamente parados e as rotas já foram descobertas e se encontram estáveis. Se os nós começam a se mover mais rapidamente, o *overhead* de roteamento aumenta unicamente o suficiente para responder corretamente à demanda das rotas atualmente em uso. Vale ressaltar ainda que o DSR, pela maneira como foi implementado o seu mecanismo de descoberta e manutenção de rotas, suporta o uso de enlaces unidirecionais e rotas assimétricas. Entretanto, segundo Johnson and Maltz [Johnson and Maltz, 1996], muitos dos principais protocolos de acesso ao meio para redes sem fio, tais como IEEE 802.11, MACA e MACAW, requerem o uso de enlaces bidirecionais, para a troca (opcional para o IEEE 802.11) de pacotes RTS (Ready To Send) e CTS (Clear To Send), para resolver o problema do terminal escondido e, no caso do IEEE 802.11, este requer ainda ACKs da camada enlace. Quando usado em cima de protocolos de acesso ao meio como estes, o DSR pode

aproveitar esta informação e usá-la para otimizações, tais como sabendo que se existe uma rota da fonte para o destino, pode-se concluir que também existe uma do destino para a fonte.

Detalhes do Funcionamento do DSR

Descoberta de Rota

Quando um nodo fonte O quer enviar um pacote a um nodo destino D, ele procura a melhor rota em um cache de rotas, denominado “Route Cache", que armazena rotas anteriormente apreendidas, seja pelo processo de descoberta de rotas anteriores, seja por escuta em modo promíscuo. Este cache é continuamente atualizado e, se uma entrada não for usada durante um certo tempo (RouteCacheTimeout), ela será apagada. No caso de escuta em modo promíscuo, se desabilita a filtragem de pacotes e se faz com que o protocolo de rede receba todos os pacotes que chegam a sua interface de rede. Destes pacotes, pode-se apreender informações potencialmente úteis, sem aumentar a sobrecarga de roteamento, o que é muito interessante, devido à escassez de banda. A quantidade de informação apreendida depende, logicamente, de os enlaces serem unidirecionais ou bidirecionais. Se a fonte previamente já havia apreendido o caminho para o destino, então, para encaminhar-lhe seus pacotes, basta colocar todo os nós do caminho de O até D no cabeçalho dos seus pacotes (roteamento por fonte). Mas, se a fonte ainda não conhece nenhuma rota que chegue até o destino, então se faz necessário efetuar o procedimento de descoberta de rota, que se especifica a seguir, com a ajuda da figura 13.

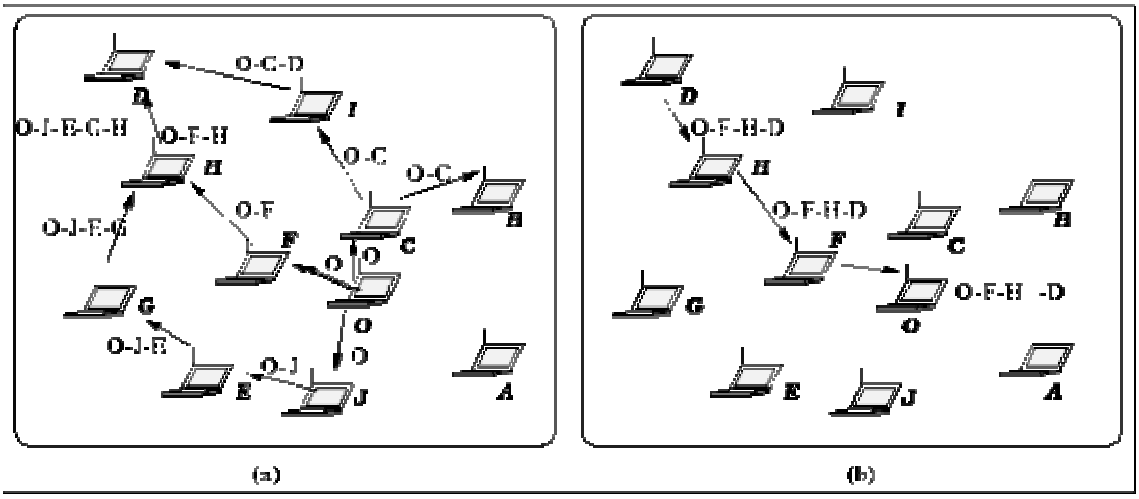


FIGURA 13 – MULTIPOINT RELAYS, [JOHNSON AND MALTZ, 1996].

Para a descoberta de rota, a fonte inunda a rede com uma requisição de rota para, dinamicamente, achar uma rota para o destino. Assim, inicialmente, a fonte envia uma mensagem, chamada Route Request, a todos os seus vizinhos J, F, C e B, na figura 13**(a),** por *broadcast*, que, caso não sejam o destino ou não tenham em seu Route Cache uma rota para o destino, repassarão a todos os seus vizinhos, sucessivamente, implementando assim a inundação da rede. Cada Route Request carrega um identificador para o nodo fonte, outro para o destino e outro para a mensagem. Além disso, carrega uma lista de todos os nós pelos quais passou até chegar ao nodo atual. Na figura 13(a), evidencia-se esta lista do caminho parcial. Mostra-se todo o caminho que o Route Request seguiria na rede, isto é, ele inundaria a rede passando por todos os nós alcançáveis desde O. Quando um nodo, que não seja o destino, recebe um Route Request, ele verifica, pelo identificador de fonte, de destino e de mensagem, se ele não transmitiu recentemente esta requisição, sendo que ele só repassará a mensagem aos vizinhos caso seja uma nova requisição (e caso não conheça uma rota para o destino). Isto é um mecanismo para se evitar sobrecarregar a rede com um número excessivo de mensagens de roteamento. Imagine que o nodo C acabou de inundar seus vizinhos com a requisição de rota que vem de O. Logo em seguida, ele recebe de um vizinho a mesma requisição de rota (sabe-se que é a mesma por causa desses três identificadores). Ora, se ele voltasse a inundar a rede com esta nova requisição, ele estaria consumindo a banda para achar caminhos piores, o que é indesejável. Caso um nodo repasse a requisição aos vizinhos, ele acrescenta a si próprio à lista do caminho parcial percorrido pelo *Route Request*. E assim, a requisição vai se espalhando pela rede. Quando a requisição finalmente chega ao destino D, este confere em seu *Route Cache* para ver se conhece um caminho para a fonte para responder-lhe com o caminho por meio de um *Route Reply*. Caso conheça previamente, é só enviar a resposta. Caso contrário, o destino começa um novo *Route Request* para descobrir um caminho até a fonte. Para evitar uma possível recursividade infinita, o *Route Reply* vai contido no *Route Request* sob a forma de *piggyback*. Este procedimento ocorre no caso geral em que existem enlaces unidirecionais. Desta forma, o DSR consegue se adaptar bem ao roteamento assimétrico, o que não se consegue com o AODV. Mas, conforme explicado anteriormente, quando o DSR se encontra em cima de protocolos de acesso ao meio como o IEEE 802.11, o MACA e o MACAW, que requerem enlaces bidirecionais, o roteamento torna-se bem mais simples e o destino pode enviar a resposta à fonte usando o caminho reverso do *Route Request*. Como alternativa, quando um nodo intermediário recebe um *Route Request* e percebe que conhece um caminho até o destino, se nesse caminho não houver nenhum nodo "duplicado", um nodo que já constava do caminho parcial no *Route Request*, então este nodo não precisa

repassar a requisição para seus vizinhos. Basta concatenar este caminho parcial com o seu caminho até o destino e enviar um *Route Reply* para a fonte, como explicado acima.

O DSR permite ao mecanismo de descoberta de rotas usar um limite de saltos para limitar o alcance da inundação de *Route Requests*. Este limite é implementado com o campo TTL (*Time To Live*) do IP. Assim, a cada nodo por que passa o *Route Request*, o valor do TTL é decrementado de uma unidade. Se este valor chegar a zero antes de chegar ao destino, o pacote é descartado. Usando este limite, pode-se implementar alguns algoritmos diferentes para a inundação. Um exemplo seria, primeiro, enviar um non-propagating *Route Request* com TTL valendo um. Assim, pode-se verificar, inicialmente, se o destino é um vizinho ou se um vizinho sabe uma rota para o destino (neste caso o *Route Cache* do vizinho é quase uma extensão do cache da fonte). Se com este mecanismo inicial se conseguir uma rota para o destino, ter-se-á uma grande economia de banda, por se evitar uma inundação de mensagens. Caso não se obtenha a rota, far-se-á a inundação da rede. Uma outra possibilidade seria implementar um mecanismo de "anel expansivo". Nele, primeiramente, tenta-se um *Route Request* com TTL igual a um. Se não se conseguir uma rota, tenta-se novamente com TTL igual a dois e, assim, vai dobrando o TTL, sucessivamente, até se conseguir uma rota. Mas, com isto aumenta a latência média para se descobrir uma rota.

Um pacote, após haver desencadeado o mecanismo de descoberta de rotas, é guardado em um *buffer,* chamado de *Send Buffer*, enquanto espera por uma rota até o destino. Enquanto ali espera, este pacote deve estar programado para, de tempos em tempos, desencadear um novo processo de descoberta de rotas. Assim, pode-se implementar um mecanismo de exponential *back-off* para o tempo entre sucessivas tentativas de descobertas de rota. Depois de um certo *timeout*, o pacote é descartado do *Send Buffer*, uma vez que pode ocorrer que o nodo destino se encontre em uma partição da rede inalcançável a partir do nodo origem.

Um outro mecanismo previsto é o de "*gratuitous*" *Route Reply*. Se um nodo **H,** por exemplo, ao fazer a escuta em modo promíscuo, escuta um pacote sendo encaminhado da fonte até o destino, de tal maneira que **H** faz parte do caminho pelo qual ainda falta ao pacote passar, mas **H** não é o próximo nodo do caminho, fica evidente a existência de um possível atalho. **H** deve então enviar à fonte um "*gratuitous*" *Route Reply* com o caminho da fonte até o nodo de quem **H** escutou estar concatenado com o resto do caminho de **H** até o destino.

Manutenção de Rota

Quando pacotes estão sendo enviados da fonte até o destino, cada nodo ao longo deste caminho é responsável por verificar se o pacote foi recebido pelo próximo nó. Assim, no caminho O, F, H, D da figura 13, O é responsável pelo recebimento de F, F é responsável pelo recebimento de H, assim como H é responsável pelo recebimento de D. Um nodo deve retransmitir o pacote através de um número máximo de tentativas até que se obtenha a confirmação de recebimento. Em muitos casos, este mecanismo de confirmação é facilmente implementado. Por exemplo, se o DSR estiver em cima do IEEE 802.11, pode-se aproveitar os ACKs da camada de enlace, definidos por este protocolo. Pode-se imaginar um ACK passivo também, no qual O sabe que F recebeu o pacote, pois escutou a transmissão de F para o próximo nó. Se estes mecanismos não estiverem disponíveis, serão necessários ACKs para cada enlace implementado, especificamente pelo DSR, o que consumirá mais banda.

Se um nodo não recebe a confirmação do próximo nodo, após um número máximo de tentativas, ele interpreta que o enlace caiu. Se este nodo for o nodo fonte, ele confere em seu *Route Cache* se conhece um outro caminho. Se não conhecer, começa uma nova descoberta de rota para o destino. Além disso, retira este caminho inválido de seu *Route Cache*, bem como outras rotas que usem este enlace. Se o nodo que constatar a queda de um enlace for um nodo intermediário, então ele envia uma mensagem *Route Error* para a fonte, através de uma rota contida em seu cache ou proveniente de uma nova descoberta de rota (caso não possua rota até a fonte no cache e não se possa garantir que os enlaces são todos bidirecionais, de forma a permitir o uso da rota reversa deste nodo até a fonte). Nesta mensagem de *Route Error* identifica-se o enlace que não é mais válido, permitindo à fonte remover este enlace de seu cache e, truncar neste ponto todas as rotas contendo este enlace. Se a fonte não conhecer um caminho alternativo, ela terá que iniciar, novamente, uma descoberta de rota. Além disso, este nodo intermediário, que enviou o *Route Error,* deve consultar o seu *Route Cache* e ver se conhece uma outra rota até o destino, pois, caso conheça, ele deve efetuar um *packet salvaging*, isto é, ele deve enviar esta mensagem até o destino através da sua rota, em vez de simplesmente descartá-la. Além disso, ele deve incrementar um contador referente ao número de vezes em que um pacote foi salvo, para evitar loops. Vale mencionar, também, que, após o recebimento de uma mensagem de *Route Error*, caso se faça necessário começar uma nova descoberta de rotas, este novo *Route Request* leva consigo o *Route Error*. Assim, os outros nós podem atualizar seu *Route Cachê,* retirando as entradas que usam o enlace que caiu, o que evita que a fonte receba como resposta novamente um caminho que use este enlace.

3.3.2. Ariadne

O Ariadne [Hu, 2002b], [Monarch, 2004] é um protocolo de roteamento reativo (*on-demand*) seguro, que não requer um hardware confiável nem exige processadores poderosos. Ele evita que os invasores mexam em rotas não comprometidas, compostas por nós não comprometidos.

Ele é baseado no *Dynamic Source Routing* (DSR – roteamento dinâmico de fonte) e depende apenas de criptografia simétrica. O Ariadne garante que o nó alvo de um processo de descoberta de rota possa autenticar o iniciador; que o iniciador possa autenticar cada nó intermediário, no caminho para o destino presente na mensagem de resposta de rota; e que nenhum nó intermediário possa remover um nó anterior na lista de nós, no pedido de rota ou nas mensagens de resposta de rota.

Este protocolo utiliza um protocolo de gerenciamento de chave chamado TESLA, que depende do sincronismo dos relógios para autenticar mensagens de roteamento. Evita, assim, que sejam injetados na rede erros de rota inválidos, forjados por qualquer nó que não seja aquele na extremidade de envio no link. O mecanismo *pre-hop hashing* (uma função hash de direção única que verifica se nenhum *hop* é omitido) também é usada pelo Ariadne.

O protocolo garante a segurança contra ataques executados por nós maliciosos que modificam ou forjam informação de roteamento. Já existe versão avançada do Ariadne que utiliza o protocolo TIK (o *TESLA* com *Instant Key Disclosure* [abertura imediata de chave]), o qual permite sincronização de tempo muito precisa entre os nós da rede, sendo imune ao ataque wormhole [Hu, 2002], [Hu, 2003b], [Hu, 2003c].

3.3.3. Ad Hoc on-demand distance vector (AODV)

O AODV foi projetado para o uso em redes *Ad Hoc* que possuam desde dezenas até milhares de nós móveis. Supõe-se que todos estes nós confiam uns nos outros. O objetivo principal do protocolo é se adaptar rápida e dinamicamente às variações das condições dos enlaces da rede, descobrindo rotas de forma a se evitar o desperdício de banda e se minimizar o uso de memória e processamento nos nós que atuam como roteadores [Perkins, 2001]. Além

disso, o AODV pode ser utilizado em cenários de baixa, média e alta mobilidade e lido com uma grande variedade de níveis de tráfego de dados.

Para garantir a economia de banda e energia, o processo de aquisição de rotas, implementado no AODV, atua sob demanda, isto é, um nó não precisa conhecer a rota até outro nó, exceto quando a comunicação entre eles seja necessária. Toda vez que um nó desconhece uma rota, válida para o destino, é necessário realizar o procedimento de descoberta de rota, o que acarreta em maior latência.

A manutenção de rotas é implementada de forma que, ao se detectar a queda de um enlace, esta informação seja propagada a todos os demais nós que usem este enlace em alguma rota ativa. Uma rota ativa é uma entrada "recente o suficiente" na tabela de roteamento com uma métrica finita. Somente rotas ativas podem ser usadas para o envio de pacotes.

O protocolo oferece duas maneiras para se detectar a queda de um enlace. A primeira utiliza um mecanismo de detecção de vizinhança, que pode ser implementada pelo próprio AODV. Neste mecanismo, mensagens de broadcast local, chamadas de *hello*, são enviadas, periodicamente, para se confirmar a conectividade local entre vizinhos. O não recebimento de uma mensagem de *hello* de um vizinho, durante um certo período de tempo, indica a queda de um enlace. A segunda maneira para se detectar a queda de um enlace é utilizar informações provenientes da camada de acesso ao meio, o que proporciona economia de banda, já que não há necessidade do envio de mensagens em *broadcast*, mesmo que este seja local. Mais detalhes sobre os processos de descoberta e manutenção de rotas serão vistos nas Seções seguintes.

O AODV, diferentemente do DSR, não utiliza o roteamento por fonte [Johnson and Maltz, 1996], o que obriga que todos os nós intermediários estabeleçam, dinamicamente, entradas em tabelas de roteamento locais para cada destino. Em compensação, não é necessário que cada pacote contenha todo o caminho da fonte até o destino, como ocorre no roteamento por fonte, o que evita a sobrecarga da rede. Além disso, a economia de banda, em relação ao roteamento pela fonte, tende a ser tão maior quanto maior for o diâmetro da rede. O AODV também faz menor uso de memória do que o DSR, já que este permite a existência de múltiplas rotas para cada destino.

Cada nó no AODV mantém um número de seqüência monotonicamente crescente, que é usado como forma de se verificar o quão recente é uma rota e, para garantir a ausência de loops de roteamento, evitando o problema da "contagem para infinito", característico dos

protocolos de vetor de distâncias. Caso ocorra a queda de um enlace, as rotas que o utilizam são invalidadas por operações que envolvem o número de seqüência e a métrica da rota.

Uma típica entrada da tabela de roteamento de um nó, que utiliza o AODV, contém os seguintes campos:

- Endereço IP do destino,
- Número de seqüência do destino,
- Indicador de validade do número de seqüência do destino,
- Indicador de validade de uma rota (válida, inválida, reparável, sendo reparada),
- Interface de rede,
- Contador de saltos (número de saltos necessários para alcançar o destino),
- Próximo salto,
- Lista de predecessores,
- Tempo de vida da rota (tempo de expiração da rota).

Quando um nó recebe uma mensagem AODV de um vizinho, ele checa a entrada da sua tabela de roteamento referente ao destino. Se não houver entrada para o destino correspondente, será criada uma nova, com o número de seqüência vindo na mensagem ou, o bit que indica a validade deste número é carregado como falso. Caso o nó já possua uma entrada para o destino em sua tabela, ela só é atualizada se o número de seqüência, presente na mensagem, for:

- Maior do que o gravado na tabela;
- Igual, mas com o (contador de saltos + 1) menor do que a métrica gravada, ou se ela desconhece o número de seqüência para este destino.

Para cada rota válida, mantida por um nó, em uma das entradas da sua tabela de roteamento, deve haver uma lista de predecessores, que podem encaminhar pacotes neste caminho. Estes nós predecessores irão receber notificações do nó, quando este detectar a quebra de um enlace para o próximo salto. A lista de predecessores contém os vizinhos para os quais se deve gerar ou encaminhar uma resposta à requisição de rota.

3.3.4. Routing on-demand acyclic multi-path (ROAM)

O protocolo de roteamento ROAM [Raju, 1999] usa coordenação internodal com *directed acyclic subgraphs*, que são derivados da distância dos roteadores até o destino. Essa operação é denominada ‘‘*diffusing computation’’*. A vantagem desse protocolo é que ele elimina o problema de busca infinita, presente em alguns dos protocolos de roteamento reativos para parar as buscas por inundação quando o destino requisitado não está mais ao alcance. Outra vantagem é que cada roteador mantém entradas em uma tabela de rota para destinos que escoam pacotes de dados através deles (ou seja, o roteador é um nodo que completa ou conecta um roteador ao destino). Isso reduz uma significativa quantia de espaço de armazenamento e largura de banda, necessários para manter uma tabela de roteamento atualizada. Outra inovação do ROAM é que cada vez que a distância entre um roteador e o destino muda por mais do que um threashold (limiar) definido, ele transmite mensagens de atualização para seus nós vizinhos, como descrito anteriormente. Ainda que isso apresente o benefício de aumentar a conectividade da rede, nas redes altamente dinâmicas isso pode impedir que os nós entrem no modo sleep (espera) para conservar energia.

3.3.5. Light-weight mobile routing (LMR)

O protocolo LMR [Corson And Ephremides, 1995] é outro protocolo de roteamento reativo, que usa uma técnica de inundação (*flooding*) para determinar suas rotas. Os nós no LMR mantêm múltiplas rotas para cada destino requisitado. Isso aumenta a confiabilidade do protocolo por permitir que os nós selecionem a próxima rota disponível para um destino específico sem iniciar um procedimento de descoberta de rota. Outra vantagem deste protocolo é que cada nodo mantém informação de rota apenas para seus vizinhos. Isso evita atrasos extras e *overheads* de armazenamento, associados com a manutenção de rotas completas. Entretanto, o LMR pode produzir rotas inválidas temporárias, que introduzem atrasos extras na determinação de um *loop* correto.

3.3.6. Temporally ordered routing algorithm (TORA)

O protocolo de roteamento TORA [Royer and Toh, 1999] está baseado no protocolo LMR. Ele usa reversão de link (*link reversal*) e procedimento de reparo de rota da mesma forma que o LMR, e também a criação de um DAGs, que é similar ao *query/reply process* usado no LMR [Corson And Ephremides, 1995]. Portanto, ele também possui os mesmos benefícios que o LMR. A grande vantagem do TORA é que ele reduz as mensagens de controle de longo alcance (*far-reaching control messages*) ao conjunto de nós vizinhos onde a mudança de topologia ocorreu. Outra vantagem do TORA é que ele também possibilita *multicasting*, ainda que esse procedimento não esteja incorporado em suas operações básicas. O TORA pode ser utilizado em associação com o *light-weight adaptive multicast algorithm* (LAM) para prover *multicasting*. A desvantagem do TORA é que o algoritmo também pode produzir rotas inválidas temporárias como no LMR.

3.3.7. Associativity-based routing (ABR)

O ABR [Royer and Toh, 1999] é outro protocolo de roteamento iniciado na fonte (*source initiated routing protocol),* que também utiliza uma técnica de *query-reply* para determinar rotas para os destinos requisitados. Entretanto, no ABR, a seleção de rota é baseada principalmente na estabilidade. Para selecionar rotas estáveis, cada nodo mantém um contador de associatividade com seus vizinhos e os links com grau de associatividade mais altos são selecionados preferencialmente em relação àqueles com grau de associatividade mais baixo. Entretanto, pode não levar ao caminho mais curto para o destino, pois as rotas tendem a durar mais. Apesar disso, menos reconstruções de rotas são necessárias e mais largura de banda estará disponível para a transmissão de dados. A primeira desvantagem do ABR é que ele requer *beaconing* periódica para determinar o grau de associatividade dos links. Essa necessidade de *beaconing* (sinalização) requer que todos os nós estejam ativos durante todo o tempo, o que pode resultar em consumo adicional de energia. Outra desvantagem é que ele não mantém múltiplas rotas ou um cache de rotas, ou seja, as rotas alternativas não estarão imediatamente disponíveis, sendo que uma descoberta de rota será requisitada a partir da falha de link. Entretanto, o ABR compensou, em certo grau, a ausência de múltiplas rotas ao iniciar um procedimento de descoberta de rotas localizadas (*localised route discovery procedure* - LBQ).

3.3.8. Signal stability adaptive (SSA)

O SSA [Dube And Tripathi, 1997] descende do ABR. Entretanto, o SSA seleciona rotas baseado na força do sinal e estabilidade de localização, ao invés de utilizar um contador de associatividade. Como no ABR, as rotas selecionadas no SSA podem não resultar no caminho mais curto para o destino. Contudo, elas tendem a ter vida mais longa, o que significa que um número menor de reconstruções de rotas será necessário. Uma desvantagem do SSA, em comparação com o DSR e o AODV, é que os nós intermediários não podem responder a pedidos de rede, enviados em direção ao destino, o que pode potencialmente criar longos atrasos antes que uma rota possa ser descoberta. Isso porque o destino é responsável pela seleção da rota para transferência de dados. Outra desvantagem do SSA é que ele não faz nenhuma tentativa de reparação de rotas no ponto em que a falha de link ocorre (como um LBQ no ABR). No SSA a reconstrução acontece na fonte. Isso pode introduzir atrasos extras, já que a fonte precisa ser notificada sobre o link quebrado antes que outro link possa ser encontrado.

3.3.9. Relative distance micro-discovery Ad Hoc routing (RDMAR)

O RDMR [Aggelou and Tafazolli, 1999] busca minimizar os overheads de roteamento calculando a distância entre a fonte e o destino, limitando cada pacote de pedido de rota (*route request*) a um certo número de *hops* (como já descrevemos). Isso significa que o procedimento de descoberta de rota pode ser conectado à região localizada (ou seja, não terá uma influência global). O RDMR também utiliza a mesma técnica quando ocorrem falhas de link (ou seja, manutenção de rota). Desta forma, conserva uma quantia significativa de largura de banda e energia de bateria. Outra vantagem do RDMR é que ele não requer uma *location aided technology* (como o GPS) para determinar os padrões de roteamento. Entretanto, o procedimento de *relative-distance micro-discovery* (micro descoberta de distância relativa) pode ser aplicado apenas se a fonte e os destinos tiverem se comunicado anteriormente. Se nenhum registro de comunicação prévia estiver disponível para fonte e destino específicos, então o protocolo se comportará da mesma forma que os algoritmos de *flooding* (ou seja, a descoberta de rota terá influência global).

3.3.10. Location-aided routing (LAR)

O LAR, baseado em algoritmos de inundação (como o DSR), reduz os *overheads* de roteamento presentes neste tipo de roteamentos. O LAR envia a requisição de informações sobre a localização do nó. O destino ao receber a informação, retorna ao nó origem sua localização. Esse protocolo assume que cada nodo conhece sua localização através de um GPS.

Para Young-Bae [Young-Bae, 1998]**,** foram propostas duas variações para o LAR. A primeira variação (LAR1) usa uma zona de requisição em formato retangular, que pode conter o destino. Cada nó, ao receber um pacote, verifica se está dentro da zona de requisição. Se estiver, envia novamente a mensagem para seus vizinhos; caso contrário, ignora os pacotes.

A segunda variação (LAR2) considera que uma mensagem possui duas informações usadas na escolha de uma zona: as coordenadas do nó e uma estimativa da distância que o nó pode estar desta coordenada. As informações são usadas pelos nós intermediários para determinar a possível zona onde o móvel destino pode se encontrar. Cada nó, ao receber uma mensagem, verifica se está mais próximo do nó do que seu vizinho; caso esteja, faz um *broadcast* para seus vizinhos, senão ignora o pacote.

3.3.11. Ant-colony-based routing algorithm (ARA)

O ARA [Günes, 2002] procura reduzir overheads de roteamento adotando o comportamento de busca por alimentos utilizado pelas formigas. Quando as formigas procuram por comida, elas começam a busca a partir de seus ninhos em direção a comida, enquanto deixam para trás um rastro temporário, chamado pheromone. Isso indica o caminho que foi tomado pela formiga e permite que outras o sigam até que o pheromone desapareça. Da mesma forma que o AODV e o DSR, o ARA é também composto por duas fases (descoberta de rota e manutenção de rota). Durante a descoberta de rota, um Forwarding ANT (FANT) é propagado através da rede (como para um RREQ). A cada *hop* (salto), cada nodo calcula um valor de pheromone, dependendo de quantos números de *hops* o FANT levou para alcançá-los. Os nós então remetem o FANT para seus vizinhos. Uma vez que o destino é alcançado, ele cria um *Backward* ANT (BANT), e o manda de volta para sua fonte. Quando a fonte recebe o BANT do

nodo destino, um caminho é determinado e a disseminação do pacote de dados começa. Para manter cada rota, cada vez que um pacote de dados viaja entre nós intermediários, o valor de pheromone é aumentado. Se não ocorrer viagem de pacotes, o valor de pheromone diminuirá com o passar do tempo até expirar. Para reparar um link quebrado, os nós primeiramente checam sua tabela de roteamento e, se nenhuma rota for encontrada, eles informam seus vizinhos para uma rota alternativa. Se os vizinhos possuem uma rota, eles informam através de *backtracking* (seguindo a ação reversa). Se o nodo fonte é alcançado e nenhuma rota foi encontrada, um novo processo de descoberta de rota será iniciado. A vantagem dessa estratégia é que o tamanho de cada FANT e BANT é pequeno, o que significa que a quantidade de *overhead* por pacote de controle introduzido na rede é minimizada. Entretanto, o processo de descoberta de rota é baseado em *flooding*, o que significa que o protocolo pode ter problemas de escalabilidade à medida que o número de nodos e fluxos crescerem na rede.

3.3.12. Flow oriented routing protocol (FORP)

O FORP [Su and Gerla, 1999] tenta reduzir o efeito da falha de link devido a mobilidade durante a transmissão de dados, prevendo quando uma rota vai ser rompida e, portanto, utilizando um link alternativo antes do acontecimento da falha da rota. Para isso, quando um nodo requer uma rota para um destino particular e a rota ainda não está disponível, uma mensagem *Flow*_REQ é transmitida através da rede, de forma similar a do pedido de rota (Route Request) no DSR. Entretanto, no FORP, cada nodo que recebe uma Flow_REQ calcula um Link Expiration Time (LET – tempo de expiração da rota) com o *hop* (salto) anterior (usando um GPS) e junta esse valor ao pacote *Flow* REQ que será, então, retransmitido por *broadcast*. Quando um pacote *Flow*_REQ alcança o destino, um *Route Expiration Time* (RET) é calculado, utilizando o mínimo de todos os LETs para cada nodo na rota, e um pacote *Flow*_SETUP é enviado de volta em direção à fonte. Durante a transmissão de dados, cada nodo intermediário adiciona o seu LET ao pacote de dados. Isso permite que o destino preveja quando uma falha do link poderá ocorrer. Quando o destino determina que uma rota está para expirar, uma mensagem *Flow*_HANDOFF é gerada e propagada via *flooding* (semelhante ao *Flow*_REQ). Portanto, quando a fonte recebe uma mensagem *Flow*_HANDOFF, pode determinar a melhor rota para realizar o *handoff* do fluxo baseada na informação dada (como o RET e *hop count* etc) no pacote *Flow*_HANDOFF. A fonte, então, envia uma mensagem *Flow*_SETUP ao longo da rota recentemente escolhida. A vantagem dessa estratégia, se comparada a outros protocolos de

roteamento *on-demand,* descritos até agora, é que ela minimiza os rompimentos das sessões em tempo real, devidos à mobilidade, através de tentativas de manutenção de fluxo constante de dados. Entretanto, por ser baseado em puro *flooding*, o protocolo pode enfrentar problemas de escalabilidade em grandes redes.

3.3.13. Cluster-based routing protocol (CBRP)

Diferente dos protocolos de roteamento *on-demand,* descritos até agora, no CBRP [Mingliang, 1999] os nós são organizados em uma hierarquia. Como a maioria dos protocolos hierárquicos descritos na seção anterior, os nós no CBRP são agrupados em *clusters*. Cada *cluster* tem um *cluster-head*, que coordena a transmissão de dados no interior do *cluster* e para outros *clusters*. A vantagem do CBRP é que apenas os *cluster heads* trocam informação de roteamento. Portanto, o número de *overhead* de controle transmitido através da rede é muito menor do que nos tradicionais métodos *flooding*. Entretanto, como em qualquer outro protocolo de roteamento hierárquico, existem *overheads* associados à formação e manutenção de *cluster*. O protocolo também enfrenta *loops* temporários de roteamento. Isso porque alguns nós podem carregar informação de topologia inconsistente devido a atrasos de longa propagação.

3.3.14. Resumo de roteamento reativo

A seguir, apresentaremos uma tabela demonstrativa das características dos protocolos reativos.

Protocolo	Estrutura de Rotemento	Multiplas Rota	Balizador	Rota principal em	Método Métrico de Rota
AODV	Flat	Não	Sim, mens. hello	Route Table	Rescente e Caminho mais curto
DSR	Flat	Sim	Não	Route Cachê	Caminho mais curto ou próxima rota disponível na Route Cachê
ROAM	Flat	Sim	Não	Route Table	Caminho mais curto
LMR	Flat	Sim	Não	Route Table	Caminho mais curto e Próximo Disponível
TORA	Flat	Sim	Não	Route Table	Caminho mais curto e Próximo Disponível
ABR	Flat	Não	Sim	Route Table	Associativa mais forte e Caminho mais curto e (item 2)
SSA	Flat	Não	Sim	Route Table	Sinal mais forte e Estabilidade
RDMAR	Flat	Não	Não	Route Table	Distância relativa mais curta e Caminho mais curto
LAR	Flat	Sim	Não	Route Cache	Caminho mais curto

ARA	Flat	Sim	Não	Route Table	Caminho mais curto
FOPR	Flat	Não	Não	Route Table	Tempo de Expiração da Rota e Estabilidade
CBRP	Hierárquico	Não	Não	Route Table	Primeira rota disponível (first fit)

TABELA 5 – COMPARAÇÃO DAS CARACTERÍSTICAS DOS PROTOCOLOS DE ROTEAMENTO REATIVOS OU SOB-DEMANDA.

Fonte: Parte das características a serem comparadas foram extraídas de [Royer and Toh, 1999].

Protocolo	TC[RD]	TC[RM]	CC[RD]	CC[RM]	Estratégia de reconfiguração de rota
AODV	*O(2D)*	*O(2D)*	*O(2N)*	*O(2N)*	Apaga a rota e notifica a origem e repara a rota local
DSR	*O(2D)*	*O(2D)*	*O(2N)*	*O(2N)*	Apaga a rota e notifica a origem
ROAM	*O(D)*	*O(A)*	*O(|E|)*	*O(6G_A)*	Apaga a rota e Inicia uma pesquisa difundindo se um sucessor estiver disponível, senão envia uma consulta com métrica infinita.
LMR	*O(2D)*	*O(2D)*	*O(2N)*	*O(2A)*	Reversão do Link e Reparo da Rota
TORA	*O(2D)*	*O(2D)*	*O(2N)*	*O(2A)*	Reversão do Link e Reparo da Rota
ABR	*O(D+P)*	*O(B+P)*	*O(N+R)*	*O(A+R)*	Broadcast de consulta localida
SSA	*O(D+P)*	*O(B+P)*	*O(N+R)*	*O(A+R)*	Apaga a rota e então notifica a origem
RDMAR	*O(2S)*	*O(2S)*	*O(2M)*	*O(2M)*	Apaga a rota e então notifica a origem
LAR	*O(2S)*	*O(2S)*	*O(2M)*	*O(2M)*	Apaga a rota e então notifica a origem
ARA	*O(D+P)*	*O(D+P)*	*O(N+R)*	*O(A+R)*	Usa rota alternativa ou retorna até encontrar a rota
FOPR	*O(D+P)*	*O(D+P)*	*O(N+R)*	*O(A+R)*	Um Flow_HANDOFF Usado para alteranar o uso da rota
CBRP	*O(2D)*	*O(2B)*	*O(2X)*	*O(2A)*	Apaga a rota e então notifica a origem e repara a rota local
D – diâmetro da rede, N – Número de nodos na rede, A – Número de nodos afetados, B – Número de áreas afetadas, G – degrau máximo da rora, S – Diâmetro dos nodos na região localizda, M – número de nodas na região localizada,					**X – Número de cluster (cada cluster tem um claster-head), R – número de nodos que formam o caminho da rota de retorno (route replay), RREP, BAND ou Flow_SETUR, P – Diametro do caminho dirigido do RREP, BAND ou Flow_SETUP, |E|-número de extremidades da rede.**
TC – Complexidade de tempo CC – Complexidade de comunicação					**RD – Discoberta de rota RM – Manutenção de rota**

TABELA 6 – COMPARAÇÃO DAS CARACTERÍSTICAS DOS PROTOCOLOS DE ROTEAMENTO REATIVOS OU SOB-DEMANDA.

Fonte: Parte das características a serem comparadas foram extraídas de [Royer and Toh, 1999].

Protocolo	Vantagem	Desvantagem
AODV	Adaptável a topologias altamente dinâmica	Problemas de escalabilidade, delay alto e utiliza mensagens de hello
DSR	Múltipla rota e Escuta Promíscua	Problemas de escalabilidade devido a rota de origem e inundação e delay alto
ROAM	Elimina pesquisa infinita	Controle de overhead alto em local de ambientes de alta mobilidade
LMR	Múltiplas rotas	Loop temporário de rota
TORA	Múltiplas rotas	Loop temporário de rota
ABR	Rota estável	Problemas de escalabilidade
SSA	Rota estável	Problemas de escalabilidade, demora alta diante de falhas de rota e reconstrução
RDMAR	Descoberta de rota localizada	Utiliza inundação se não houver comunicação anterior entre os nodos
LAR	Descoberta de rota localizada	Baseado na rota de origem, utiliza inundação se a informação de localização não estiver disponível.
ARA	Overhead bairo e baixo controle de tamanho de pacote	Processo de descoberta de rota baseado em inundação
FOPR	Emprega técnicas de minimização de falhas de rota	Processo de descoberta de rota baseado em inundação
CBRP	Somente cluster-head trocam informações de roteamento	Gerenciamento de cluster, looping temporários

TABELA 7 – COMPARAÇÃO DAS CARACTERÍSTICAS DOS PROTOCOLOS DE ROTEAMENTO REATIVOS OU SOB-DEMANDA.

Fonte: Parte das características a serem comparadas foram extraídas de [Royer and Toh, 1999].

3.4. Protocolos de Roteamento Híbridos

Protocolos de roteamento híbridos são uma nova geração de protocolos de natureza tanto pró-ativa quanto reativa, ou seja, é uma fusão das duas formas de roteamento. Dependendo das condições da rede, uma forma de roteamento é adotada [Zhou, 2004]. Esses protocolos são planejados para aumentar a escalabilidade, uma vez que permitem que os nós com grande proximidade trabalhem juntos, para formarem uma espécie de *backbone* que reduza os *overheads* de descoberta de rota. Isso é obtido principalmente através da manutenção pró-ativa de rotas para nós atuando nas proximidades e, da determinação reativa de rotas para nós distantes, com o uso de uma estratégia de descoberta de rota. A maioria dos protocolos híbridos propostos até hoje está baseada em zona, o que significa que a rede é particionada ou vista como um número de zonas para cada nó. Outros agrupam os nós em árvores ou *clusters*. Comumente referenciados como protocolos de rotemento hibrido-balanceado (*balanced-hybrid routing*), são uma combinação do rotemento de vetor de distância (*distance-vector*), que trabalha compartilhando seu conhecimento da rede inteira com seus vizinhos e, protocolos *link-state,* onde cada nodo mantém informação de topologia sobre a rede através de trocas periódicas de mensagens de *Link-Stat.* Entretanto, usam o rotemaneto vetor-de-distância (*distance-vectors*) para métricas mais precisas, na determinação dos melhores caminhos até os nodos destinos, e para reportar mudanças de topologia na rede.

Os protocolos de roteamento híbrido são a terceira classificação de algoritmos de rotemaneto.

3.4.1. Zone routing protocol (ZRP)

O protocolo ZRP [Haas, 1998] é um híbrido entre reativo e pró-ativo e é definido baseado em zonas de roteamento. Requer que, dentro de sua zona, o nó conheça as informações de roteamento, utilizando algoritmos pró-ativos. Isto permite que as informações de roteamento sejam propagadas apenas localmente. Quando as informações são inter-zonas, o algoritmo utilizado é do tipo reativo.

A zona é definida por cada nó e inclui os nós que estão a uma distância mínima de *hops* do nó em questão. Na maioria das vezes, a distância é um número pré-definido, que é

denominado como raio da zona. Se o destino estiver dentro da zona do nó que disparou o procedimento, então o pacote é imediatamente enviado, pois o nó já conhece o caminho; senão, ele envia o pacote através de *multicast* para todos os nós que estão na borda da zona de roteamento, perguntando se conhecem uma rota para o destino. Se conhecerem, os nós mandam uma resposta afirmativa e a conexão é fechada. Caso não conheçam, eles enviam a mensagem para os nós da borda de sua zona. Este procedimento segue até que uma confirmação chegue ou que o número de *hops* alcance o máximo.

Como o ZRP é um híbrido entre algoritmos reativos e pró-ativos, ele pode retirar vantagens das duas abordagens. O roteamento pode ocorrer rapidamente em nós que estão na mesma zona, pois os caminhos já são previamente conhecidos. Além disso, o algoritmo apresenta boa escalabilidade, as mudanças na topologia são rapidamente detectadas e as tabelas de roteamento são pequenas. A desvantagem nas intra-zonas é que, como o protocolo de roteamento não é especificado, várias zonas podem ter protocolos diferentes. Outra desvantagem acontece quando ocorrem informações inter-zonas, pois o algoritmo usado é do tipo sob-demanda. Neste caso, o principal problema é o atraso decorrente da espera pela resposta de requisição de rotas.

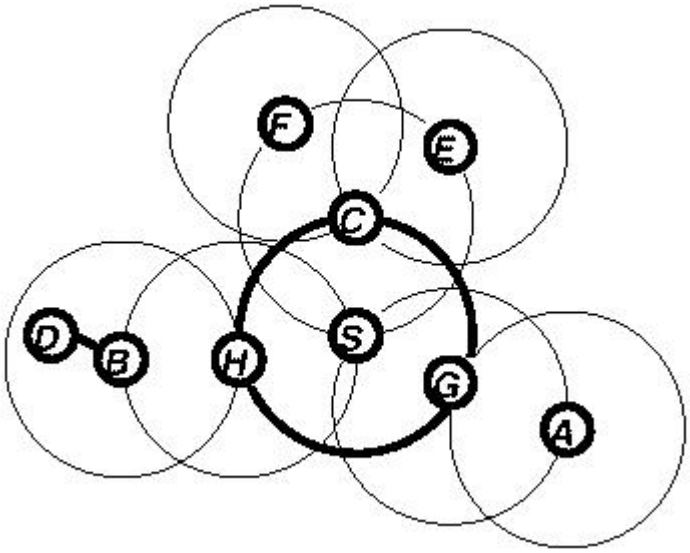


FIGURA 14 – ZONE ROUTING: DESCOBRIMENTO DE ROTA DE S PARA D, [HAAS , 1998].

3.4.2. Zone-based hierarchical link state (ZHLS)

Diferente do ZRP, o protocolo de roteamento ZHLS [Joa-ng, 1999] utiliza estrutura hierárquica. No ZHLS, a rede é dividida em zonas não-sobrepostas (*non-overlapping*), e cada

nodo tem um ID de nodo e um ID de zona, que é calculado com o uso do GPS. A topologia hierárquica é composta de dois níveis: a topologia de nível de nodo e a topologia de nível de zona. No ZHLS, o gerenciamento de localização foi simplificado. Isso porque nenhum *cluster-head* ou gerenciador de localização (*location manager)* é utilizado para coordenar a transmissão de dados. Isso significa que não existe *overhead* de processamento associado à seleção de *cluster-head* ou gerenciador de localização em comparação ao HSR, ao MMWN e aos protocolos CGSR. Isso também significa que um ponto único de falha e picos de excesso de tráfego podem ser evitados. Outra vantagem do ZHLS é que, em comparação aos protocolos reativos puros como o DSR e o AODV, ele reduz os *overheads* de comunicação. No ZHLS, quando uma rota para um destino remoto é requisitada, ou seja, o destino está em outra zona, o nodo fonte transmite um pedido de localização de nível de zona para todas as outras zonas, gerando *overhead* significativamente mais baixo do que na abordagem de *flooding* nos protocolos reativos. Outra vantagem do ZHLS é que o caminho de roteamento é adaptável à topologia variante, já que apenas o ID de nodo e o ID de zona do destino são requeridos para o roteamento. Isso significa que nenhuma busca de localização adicional é requisitada, desde que o destino não migre para outra zona. Entretanto, em protocolos reativos, qualquer quebra de *link* intermediário invalida a rota, sendo necessário iniciar outro procedimento de descoberta de rota. A desvantagem do ZHLS é que todos os nós precisam ter um mapa de zona estática pré-programado para poder funcionar. Isso pode ser impossível em aplicações onde os limites geográficos da rede são dinâmicos. No entanto, ele é altamente adaptável a topologias dinâmicas e gera bem menos *overhead* do que os protocolos reativos puros, o que significa que ele pode escalar bem em grandes redes.

Funcionamento do mapeamento da topologia ao nível de zona.

Um nodo conhece a sua localização física através de técnicas de geolocalização como GPS; então, pode determinar o ID da sua zona, traçando sua localização física para um mapa de zona, que tem que ser trabalhado fora, na fase de projeto. O tamanho de zona depende de fatores como mobilidade do nodo, densidade da rede, poder de transmissão e características de propagação. As partições podem ser baseadas em partições geográficas simples ou em partições de propagação de rádio. A partição geográfica é muito mais simples e não requer qualquer medida de características de propagação de rádio, considerando que a partição de propagação de rádio é mais precisa para freqüência reutilizada. A partição de propagação de Rádio é preferível

se uma medida de propagação pode ser feita na fase de projeto. Embora algumas aplicações, como operações de salvamento em desastres e comunicação militar tática, não permitam tais medidas, uma partição geográfica deverá ser usada.

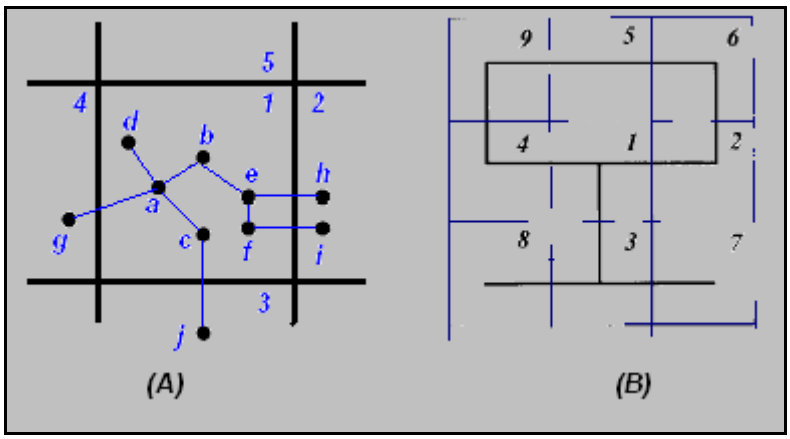


FIGURA 15 – (A) – TOPOLOGIA AO NÍVEL DE NODO E (B) AO NÍVEL DE ZONA, [JOA-NG, 1999].

Dois são os níveis de topologia definidos em ZHLS: A topologia em nível de nodo e a topologia em nível de zona, que ocorre se qualquer dos nodos estiver dentro da gama de comunicação, se uma ligação física existe. Como os nodos estão juntos, conectados por estas ligações físicas, a topologia em nível de nodo (figura 15(A)) provê a informação. Por exemplo, na figura 15, se o nodo *(a)* quer enviar um pacote de dados para o nodo *(f)*, os dados têm que atravessar por *(a→b→e→f)* -- -Se houver uma ligação física que conecte duas zonas quaisquer, duas zonas pelo menos, uma ligação virtual existe então. A topologia em nível de zona (figura 15(B)) conta como as zonas estão conectadas por estas ligações virtuais. Por exemplo, na Fig. 2, as ligações virtuais entre zona 4 e zonas 3 são 4→1→3.

3.4.3. Scalable location update routing protocol (SLURP)

Da mesma forma que o ZLHS, no SLURP [Sanzgiri, 2002] os nós são organizados em um número de zonas não-sobrepostas. Entretanto, o SLURP ainda reduz o custo de manutenção da informação de roteamento através da eliminação da descoberta de rota global. Isso é obtido através da designação de uma *home region* para cada nodo na rede. A *home region* é uma zona ou região específica que é determinada utilizando-se uma função de mapeamento estático, f (ID do nó) →ID da região, onde f é uma função *many-to-one* que é estática e

conhecida por todos os nós. Um exemplo de uma função que pode executar o mapeamento de zona estático é f(ID do nó) = g(ID do nó) modK, onde g(ID do nó) é uma *random number generating function* (função de geração de número aleatório) que usa o ID de nodo como entrada e dá como saída um número alto, e k é o número total de *home regions* na rede. Então, uma vez que o Id de nodo de cada nodo é constante (ou seja, um endereço MAC), a função irá calcular sempre a mesma *home region*. Portanto, todos os nós podem determinar a *home region* utilizando esta função, desde que possuam seu ID de nó. Cada nodo mantém sua localização atual (zona atual) com a *home region* através de envio por *unicasting* de uma mensagem de atualização de localização em direção à *home region*. Assim que o pacote de atualização de localização atinge a *home region*, ele é transmitido por *broadcast* para todos os nós na *home region*. Consequentemente, para determinar a localização atual de qualquer nó, cada nodo pode enviar, por *unicast,* um pacote de descoberta de localização para a *home region* dos nós requisitados (ou a área ao redor da *home region*) com o intuito de encontrar sua localização atual. Uma vez que a localização é encontrada, a fonte pode começar a enviar dados em direção ao destino utilizando *most forward with fixed radius* (MFR) geographical forwarding algorithm. Quando um pacote de dados atinge a região onde se encontra o destino, então o **roteamento de fonte** é utilizado para levar o pacote de dados até o destino. A desvantagem do SLURP é que ele também se apoia em um mapa estático de zona pré-programado (da mesma forma que o ZHLS).

3.4.4. Distributed spanning trees based routing protocol (DST)

No DST [Johnson and Maltz, 1996] os nós na rede estão agrupados em um número de árvores. Cada árvore tem dois tipos de nós: nodo de rota e nodo interno. O nodo raiz controla a estrutura da árvore e a possibilidade de que ele se misture a outra árvore, e o restante dos nós, em cada árvore, são os nós regulares. Cada nodo pode estar em três estados diferentes: *router, merge* e configure, dependendo do tipo de tarefa que ele está tentando executar. Para determinar a rota, o DST propõe duas estratégias de roteamento diferentes: *hybrid tree-flooding* (HFT) e distributed spanning tree shuttling (DST). Na HTF, os pacotes de controle são enviados para todos os vizinhos e pontes adjacentes, na árvore *spanning*, onde cada pacote é mantido por um período de tempo chamado *holding time*. A idéia do *holding time* é que, como a conectividade aumenta e a rede torna-se mais estável, ele pode ser útil para o *buffer* e pacotes de rota quando a conectividade aumenta com o passar do tempo. No DST, os pacotes de controle são

disseminados a partir da fonte e retransmitidos por *broadcast* ao longo das bordas da árvore. Quando um controle atinge o nível de um nodo folha, ele é enviado de volta pela árvore até que alcance uma certa altura, denominada *shuttling level*. Quando o *shuttling level* é atingido, o pacote controle pode ser enviado novamente descendo pela árvore ou pelas pontes adjacentes. A principal desvantagem do algoritmo DST é que ele se apoia no nodo raiz para configurar a árvore, criando um único ponto de falha. Além disso, o *holding time* usado para o *buffer* dos pacotes pode introduzir atrasos extras na rede.

3.4.5. Distributed dynamic routing (DDR)

O DDR [Nikaein and Laboid, 2001], [Nikaein and Laboid, 2002] é também um protocolo de roteamento baseado em árvore. Entretanto, ao contrário do DST, nos DDR as árvores não requerem um nodo raiz. Nessa estratégia, as árvores são construídas utilizando periódicas mensagens de *beaconing* (sinalização), que são trocadas apenas entre os nós vizinhos. As árvores na rede formam uma floresta que é conectada como um todo via nós gateway (ou seja, nós que estão em uma gama de transmissão, mas pertencem a árvores diferentes). Cada árvore na floresta é formada por uma zona que é designada por um ID de zona, através da execução de um algoritmo de nomeação de zona (*zone naming algorithm*). Além disso, uma vez que cada nodo pode pertencer a uma única zona apenas (ou árvore), então a rede pode ser também vista como diversas zonas não-sobrepostas. O algoritmo DDR consiste em seis fases: eleição do vizinho preferido, construção da floresta, *clustering* intra-zona (intra-árvore), *clustering* inter-zona (inter-árvore), nomeação da zona e particionamento da zona. Cada uma dessas fases é executada baseada na informação recebida nas mensagens de *beaconing* (sinalização). Durante a fase de inicialização, cada nodo começa a fase da eleição do vizinho preferido. O vizinho preferido de um nodo é o nodo que possui o maior número de vizinhos. Após isto, uma floresta é construída através da conexão de cada nodo ao vizinho preferido. A seguir, o algoritmo de *intra-tree clustering* é inicializado para determinar a estrutura da zona (ou da árvore) e para construir a tabela de roteamento intra-zona. Segue-se a execução do algoritmo *inter-tree* para determinar a conectividade com as zonas vizinhas. Cada zona é nomeada zona através da execução de um algoritmo de nomeação de zona (*zone naming algorithm*) e a rede é particionada em diversas zonas não-sobrepostas. Para determinar rotas, protocolos de roteamento híbrido *Ad Hoc* (HARP) [Nikaein and Bonnet, 2001] trabalham em cima do DDR. Os HARP

usam as tabelas de roteamento intra-zona e inter-zona, criadas pelo DDR, para determinar um caminho estável entre a fonte e o destino. A vantagem do DDR é que, ao contrário do ZHLS, ele não se apoia em um mapa estático da zona para executar roteamento e ele não requer um nodo raiz ou um *clusterhead* para coordenar dados e controlar transmissão de pacotes entre diferentes nós e zonas. Entretanto, os nós que foram selecionados como vizinhos preferidos podem tornar-se picos de excesso de performance porque eles transmitiriam mais roteamentos e pacotes de dados do que todos os outros modelos, ou seja, esses nós requereriam mais recarregamento na medida em que teriam menos *sleep time* do que os outros nós. Além disso, se um nodo é o vizinho preferido para muitos de seus vizinhos, muitos nós podem querer se comunicar com ele. Isso significa que a contenção de canais aumentaria em torno do vizinho preferido, o que resultaria em atrasos maiores experienciados por todos os nós vizinhos antes que eles pudessem reservar o meio. Em redes com alto tráfego, isto pode resultar em redução significativa da taxa de transferência, com a redução de pacotes entregues quando os *buffers* estioverem cheios.

3.4.6. Resumo do roteamento híbrido

Os protocolos de roteamento híbridos possuem potencial para proporcionar uma escalabilidade mais alta do que os protocolos reativos ou pró-ativos puros. Isso porque eles tentam minimizar o número de nós, retransmitindo por *broadcast,* através da definição de uma estrutura, que permite aos nós trabalharem juntos, organizando como o roteamento será executado. Através desse trabalho conjunto, os melhores ou mais adequados nós podem ser utilizados para executar descoberta de rotas. Outra inovação dos protocolos de roteamento híbridos é que eles tentam eliminar pontos únicos de falha e nós com obstáculos na rede. Isso é obtido permitindo-se que qualquer número de nós execute roteamento ou remessa de dados se o caminho preferido deixa de ser disponibilizado.

Características	Protocolos				
	ZRP	ZHLS	SLURP	DST	DDR
Estrutura de roteamento	Flat	Hierarquico	Hierarquico	Hierarquico	Hierarquico
Mutiplas rotas	Não	Sim, se exitstir mais que um link virtual.	Sim, dependendo se o nodo principal for encontrado pelo MFR.	Sim, se disponível	Sim, se houver nodos gateway disponíveis
Utiliza Mensagem de sinalização	Sim	Não	Não	Não	Sim

Método de métrica de rota	Menor Caminho	Menor Caminho ou próximo link virtual disponível	MFR para remeter entre zonas	Remete usando os vizinhos de árvore e o transporte usando bridge	Roteamento Estável
Manutenção de rotas	Inter-zona e tabelas de inter-zona	Inter-zona e tabelas de inter- zona	Cachê de localização a um nodo_lista	Tabela de rotas	Inter-zona e tabelas de inter- zona
Estratégia de reconfiguração de rota	Conserto de rota a ponto de fracasso e Fonte Notificação (A fonte pode ou pode não ser notificada.)	Local pedido (Uma requisição de localização será enviado se a zona ID de um nodo mudar)	Notificação de fonte, então localização discoberta	Holding Time ou shuttling	Notificação de fonte, então localização discoberta
Complexidade de tempo (Descuberta de rota)	Intra: O(I)/Inter O(2D)	Intra: O(I)/Inter O(2)	Intra:$O(2Z_D)$/ Inter O(2D) (no cenário de pior caso, o nodo de fonte e a região home do destino está nas extremidades opostas da rede.)	Intra: $O(Z_D)$/Inter O(D)	Intra: O(I)/Inter O(2D)
Complexidade de tempo (manutenção da rota)	O(I)/ O(2D)	O(I)/O(2)	$O(2Z_D)$/ O(2D)	$O(Z_D)$/O(D)	O(I)/O(2D)
Complexidade de comunicação (Descuberta de rota)	$O(Z_N)$/ O(N+V)	O(N/M)/O(N+V)	O(2N/M)/O(2Y)	$O(Z_N)$/O(N)	$O(Z_N)$/O(N+V)
Complexidade de comunicação (manutenção da rota)	$O(Z_N)$/O(N+V)	O(N/M)/O(N+V)	O(2N/M)/O(2Y)	$O(Z_N)$/O(N)	$O(Z_N)$/O(N+V)

Z_D = Diâmetro do cluster; I = intervalo de atualização inter-zonas; N=Número de nodos na rede; M=Número de zonas ou clustes na rede; Z_N = Número de nodos na zonas, clusters ou árvores; Y= Número de nodos no caminho para a região home; V= Número de nodos no caminho da rota de replay; SPF = único ponto de falha;

Protocolos	Vantagens	Desvantagens
ZRP	Retransmissões reduzidas	Overlapping zones (zonas de sobreposição)
ZHLS	único ponte de falha reduzidos, baixo controle de overhead	Requer mapa de zonas estático
SLURP	Utiliza Regiões home para discoberta de localização	Requer mapa de zonas estático
DST	Retransmissões reduzidas	Nodo raiz
DDR	Nenhum mapa de zona ou coordenador de zona. Não requer um nodo raiz ou um clusterhead para coordenar dados e controlar transmissão de pacotes entre diferentes nós e zonas	Vizinhos preferências podem ser tornar gargalos em

TABELA 8 – COMPARAÇÃO DAS CARACTERÍSTICAS DOS PROTOCOLOS DE ROTEAMENTO HÍBRIDOS.

4. Segurança em Redes MANET

Projetar protocolos de roteamento seguro para redes *MANET* é uma tarefa difícil, devido à natureza altamente dinâmica deste tipo de rede e ao alto grau de comprometimento entre os nós da rede, já que todos dependem uns dos outros para o pleno funcionamento da rede. A qualidade conseguida depende do trabalho de cada nó.

Os protocolos, atualmente utilizados, foram propostos e analisadas sem abordar os aspectos de segurança relativos ao roteamento das redes *MANET*. Assim sendo, não foram considerados os possíveis ataques que poderiam assolar esses protocolos, especialmente a utilização de nós maliciosos na rede. As vulnerabilidades ligadas às redes *MANET* são atribuídas ao grau de comprometimento entre seus membros, tornando-se uma grave falha de segurança. Conforme já afirmado anteriormente, é muito mais fácil interceptar um sinal de uma rede *Wireless* do que de uma rede cabeada, não sendo necessário ter acesso físico e nem estar no mesmo ambiente para monitorar o canal de comunicação.

A vulnerabilidade dos protocolos de roteamento pode ser classificada em dois grandes grupos: vulnerabilidade dos mecanismos básicos e vulnerabilidade dos mecanismos de segurança.

A primeira pode ser tratada por esquemas de criptografia, ou seja, os mecanismos básicos de operação da rede, onde o roteamento é o mais crítico deles, passariam a trocar informações criptografadas. A segunda é bem mais complexa e agrega um certo número de diferentes soluções, mas tornou-se consenso que o ponto crítico é o gerenciamento das chaves do sistema de segurança.

O sistema torna-se vulnerável em relação aos mecanismos básicos quando, de alguma forma, é possível injetar, modificar ou replicar informações errôneas sobre a operação da rede, ou ainda, comportar-se de forma maliciosa e não cooperativa, objetivando a interrupção da operação da rede. Com relação aos mecanismos de segurança, apresenta-se vulnerável o sistema que permite a criação, distribuição e uso de chaves de criptografia, indevidamente e maliciosamente, por entidades não autorizadas, com o objetivo de acessar, como membro autenticado, os recursos e serviços da rede.

As ameaças que intentam contra as redes *MANET* vão desde a captura do dispositivo móvel até os níveis da aplicação. Os ataques podem ser classificados como ativos ou passivos e internos ou externos. Tais ataques visam, basicamente, a descoberta de informações antes inacessíveis e o impedimento da realização dos serviços da rede. O mais severo dos ataques é o ativo interno. Nele, o nodo torna-se comprometido e realiza um ataque dito protegido, já que ele é um nodo autenticado da rede. O ataque pode, inclusive, ser realizado por vários destes nós comprometidos, operando em grupo. O menos comprometedor dos ataques, mas não menos importante, é o tipo passivo externo, que consiste na "escuta" das informações que trafegam na rede.

A ameaça de impedimento de serviço constitui um grade risco num sistema distribuído, como em uma rede *MANET,* e pode ter sua origem numa falha de operação não intencional ou em ações maliciosas de parte de elementos da rede. Suas conseqüências para a operação da rede dependem do tipo de aplicação e do poder de ataque dos malfeitores. O ataque de personificação também pode trazer grandes prejuízos aos usuários da rede, já que consiste na capacidade de um nodo se fazer passar por outro e, manter um comportamento malicioso a fim de inserir e obter informações não disponíveis para sua real identidade. A possibilidade de descoberta de uma informação por entidade não autorizada na rede deve ser amplamente combatida, principalmente em se tratando de aplicações militares e comerciais. Estas informações críticas devem ser protegidas contra ataques de exposição, a fim de manter em sigilo detalhes como localização dos nós, chaves, senhas e identidade de operadores e proprietários dos dispositivos.

4.1. Tipo de Ataques

Neste item, observaremos diferentes tipos de ataques e discutiremos ataques específicos contra roteamento de redes *MANET*. A ameaça é definida por Shirey [Shirey, 2000] como um potencial para violação de segurança, que existe quando há uma circunstância, capacidade, ação ou evento que poderia violar a segurança e causar danos. Uma ameaça existe quando um invasor possui a habilidade de tirar vantagem de uma debilidade existente na segurança.

As redes *Wireless* tornaram-se alvo de exaustivos estudos e muitos ataques foram desenvolvidos e/ou adaptados para poderem aproveitar-se das fraquezas apresentadas neste tipo de redes. As redes *Wireless* continuam apresentando falhas graves de segurança e problemas na implementação e conceituação do próprio protocolo.

A ISO (International Organization for Standardization) [ISO, 1989] define a segurança como a tentativa de se minimizar as vulnerabilidades de valores e recursos dos sistemas. Entende-se por vulnerabilidade as falhas ou falta de segurança das quais pessoas mal intencionadas podem se valer para invadir, subtrair, acessar ilegalmente, adulterar e destruir informações confidenciais, além de poder comprometer, corromper e inutilizar o sistema.

A pesquisa sobre a segurança das redes de computadores leva em consideração quatro aspectos fundamentais:

- **Confidencialidade**: objetiva prevenir a obtenção de informação não autorizada;
- **Disponibilidade**: objetiva prevenir que recursos ou informações fiquem indisponíveis;
- **Integridade**: objetiva prevenir que mudanças ocorram em informações sem autorização;
- **Usabilidade**: objetiva prevenir que um serviço tenha sua utilidade deteriorada devido à segurança.

As características acima citadas devem ser balanceadas para que: o sistema não fique tão seguro a ponto de usuários legítimos não conseguirem utilizá-lo eficientemente; e que este sistema também não seja inseguro a ponto de permitir a ação de usuários não autorizados. Um fator importante, relacionado à segurança de redes sem fio, é o fato de que os ataques gerados dentro destas redes são disparados dentro do mesmo domínio de colisão, ou seja, o atacante se comunica com o mesmo concentrador de tráfego do sistema, o qual almeja atacar.

Existem basicamente duas classes de ataques: passivos e ativos. Os ataques passivos não enviam pacotes para rede; eles somente efetuam "escutas" não autorizadas. Estes ataques normalmente são ataques contra a privacidade ou anonimato das comunicações, contra o bom funcionamento da rede e contra os protocolos de roteamento. Já os ataques ativos injetam pacotes na rede e, geralmente, também efetuam "escutas" não autorizadas.

4.1.1. Ameaça por links comprometidos

Um link está comprometido quando um invasor pode, de alguma forma, acessar um meio físico e/ou ter algum controle sobre o canal. Esta ameaça ocorre quando não existem mecanismos de controle de acesso aplicados aos canais e meios físicos, ou quando tais mecanismos podem ser burlados. O invasor pode ter acesso à comunicação, reproduzir, retardar ou excluir mensagens de roteamento, ou ainda, quebrar sessões de roteamento entre roteadores autorizados sem participar do intercâmbio de roteamento.

4.1.2. Ameaças por dispositivos comprometidos

Um dispositivo (roteador) comprometido é um roteador autorizado com *bugs* de software de roteamento, defeitos de hardware e/ou configurações incorretas/não intencionais. Essa ameaça acontece quando não existem mecanismos para verificar a integridade do sistema de um dispositivo (roteador), ou seja, o roteador está trabalhando corretamente, conforme o esperado pelo administrador de autoridade da rede, ou tais mecanismos podem ser burlados. O invasor pode reivindicar, de forma não apropriada, a autoridade sobre alguns recursos da rede ou violar protocolos de roteamento, por exemplo, repassando informações inválidas de roteamento.

4.1.3. Ameaça por dispositivos não autorizados

Um dispositivo (roteador) não autorizado participa do intercâmbio e da computação de roteamento, não estando autorizado (explícita ou implicitamente) pelo administrador de autoridade da rede. Essa ameaça acontece quando não existe mecanismo de controle de acesso, aplicado às sessões ou aos intercâmbios de roteamento, ou quando tal mecanismo pode ser burlado. O invasor pode obter conhecimento da topologia da rede através do intercâmbio de roteamento, ou ainda, fazer qualquer coisa que um roteador comprometido pode fazer.

4.1.4. Ameaça por dispositivos disfarçados

Um dispositivo (roteador) disfarçado assume ilegitimamente a identidade de outro roteador. Essa ameaça acontece quando não existem mecanismos de autenticação (origem de dados ou entidades par), ou quando tais mecanismos podem ser burlados. O invasor pode fazer qualquer coisa que um roteador autorizado faz.

4.1.5. Spoofing

O ataque do tipo *spoofing* ocorre quando um nó representa uma identidade falsa na rede, através, por exemplo, da alteração de seu endereço MAC ou IP nos pacotes em andamento. Quando um invasor é bem sucedido na realização de um *spoof*, ele faz o papel de um roteador ou nodo disfarçado.

Nos dispositivos para redes sem fio, existe a particularidade de permitir a troca do endereço físico. Com isso, atacantes mal intencionados podem capturar, através de técnicas de escuta maliciosas e espionagem, um endereço MAC válido de um cliente, trocar seu endereço pelo do cliente e utilizar a rede.

Além deste tipo de MAC *Spoffing*, existe o MAC *Spoffing* da placa de rede guiada dos *access points* (para redes infra-estruturadas). Ou seja, os *access points* são capazes de trocar seus endereços MAC das placas de redes tradicionais, burlando, assim, os *firewalls* internos à LAN.

4.1.6. Ataques de Modificação

Ataque de modificação acontece quando o tráfego é desviado para um determinado nó. O nó malicioso altera informações de mensagens de roteamento recebidas, gerando informações de rotas falsas, ou tentando atrair para si o tráfego da rede, fazendo com que todas as rotas passem por ele. Essa atitude faz com que um nó se torne atraente aos nós vizinhos,

fazendo parte de suas rotas, podendo, inclusive, atingir outros nós, através de inundações da rede, com rotas falsas.

Para os protocolos que mantém tabelas de vetores de distância, os principais ataques desse tipo são, justamente, a alteração destes vetores de distância, e os de número de seqüência [Wang, 2002], [Wang, 2002b]. No ataque de vetor de distância, o nodo malicioso informa uma métrica menor para a sua rota, fazendo com que os nodos adjacentes prefiram utilizar a rota que passa por ele. No DSR, por exemplo, isso pode ser feito diminuindo o número de nodos na lista de nodos da *source route*, diminuindo assim o número de saltos. Já no protocolo AODV, o nodo malicioso simplesmente diminui o valor do campo *Hop Count*. Em um ataque de número de seqüência, o nodo malicioso, ao receber uma mensagem de roteamento, altera o valor do número de seqüência do destino, fazendo com que todos os nodos que receberem essa rota passem a utilizá-la, já que o seu número de seqüência é muito maior, e eles a consideram, então, uma rota mais atualizada que as outras.

Nós inimigos ou comprometidos podem participar do processo de descoberta de rotas e aproveitar-se disso. Os pacotes de *route request (RREQ)* e *route reply (RREP)* podem ser alterados enquanto trafegam, ou podem ser forjados, causando diversas anomalias no funcionamento da rede [Naldurg, 2001], [Zhang, 2000] e [Dahill, 2001].

Os pacotes usados pelos protocolos de roteamento, quando sujeitos a ações como as citadas acima podem causar:

- **Rotas com loops;**
- **Timeouts demorados;**
- **Métricas falsas ou exageradas;**
- **Repetição de updates antigos/desatualizados;**
- **Levando a negação de serviços (DOS).**

Alteração Do Campo Destination Sequence Numbers

Com uma simples falsificação do *destination sequence numbers* é possível redirecionar o tráfego da rede e, até mesmo, impedi-lo de alcançar seu destino [Molva and Michiardi, 2002].

Considerando a seguinte rede *Ad Hoc*:

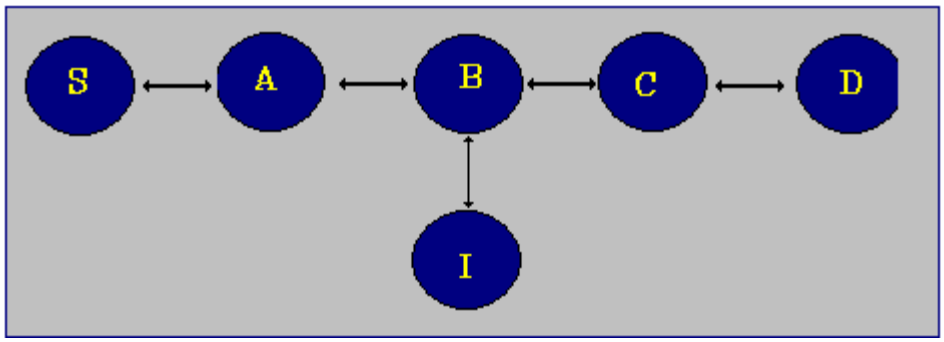


FIGURA 16 – DEMONSTRAÇÃO DO ROTEAMENTO EM UMA REDE *MANET* [MOLVA AND MICHIARDI, 2002].

Quando o nodo S deseja comunicar-se com o nodo D, ele deve iniciar o processo de descoberta de rota. Na rede que temos como exemplo, o nodo A seria o primeiro a receber o RREQ e, daí por diante, iria propagá-lo para toda a rede. Considerando o funcionamento normal da rede, num dado momento, o nodo B receberia o RREP e concluiria o processo de descoberta e o transmitiria para o nodo A, que, por sua vez, o passaria para S. Porém, se o nodo I criasse um RREP falso, contendo um *destination sequence number* maior que o do pacote verdadeiro e informando que a rota passa por ele, todo o tráfego enviado por S seria transmitido de B para I.

Alteração Da Quantidade De Saltos

Um dos parâmetros utilizados pelos protocolos de roteamento, baseados em vetor de distância para a escolha da melhor rota, é a quantidade de saltos, que podem ser facilmente alterados. Mudando a quantidade de saltos de um pacote para um número mais baixo do que o número apresentado por alguma rota, um atacante poderia desviar o tráfego, fazendo com que todos os pacotes passassem obrigatoriamente por ele. É possível também forjar seu próprio endereço IP, fazendo-se passar por outro nodo da rede, desviando ou obrigando o tráfego a passar por esse nó. A alteração do campo de endereçamento por um valor forjado é chamada de *spoofing*. A quantidade de saltos também poderia ser alterada para um valor maior do que o apresentado por uma rota válida, alterando, assim, o fluxo "correto" do tráfego. A alteração do campo com a quantidade de saltos de uma rota, para um valor maior, poderia ser combinada com a técnica de *spoofing*.

Abaixo, um exemplo de como a técnica de *spoofing,* combinada com a alteração do campo que indica a quantidade de saltos, poderia ser usada. Considerando a seguinte rede *Ad Hoc*:

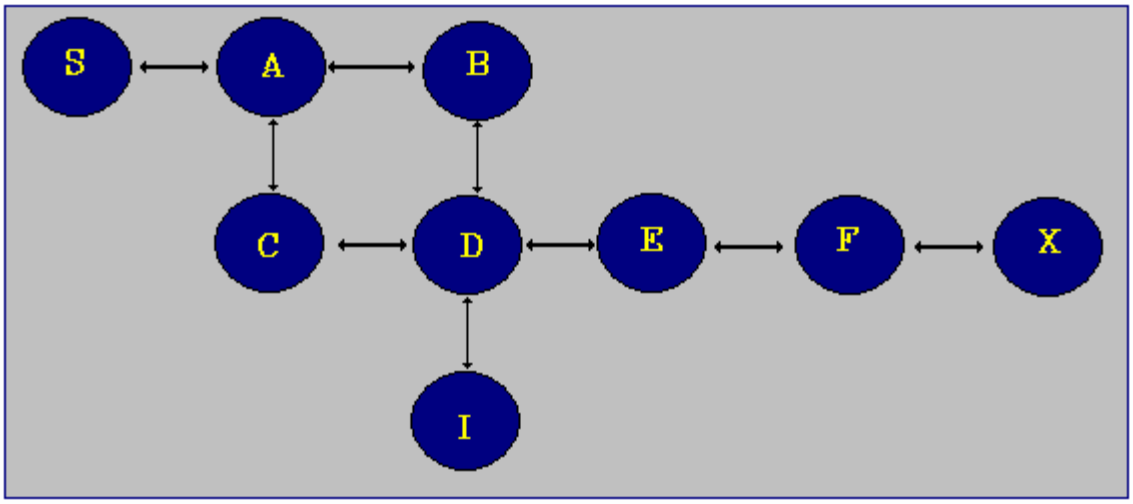


FIGURA 17 – DEMONSTRAÇÃO DO ROTEAMENTO EM UMA REDE *MANET,* [MOLVA AND MICHIARDI, 2002].

Imaginado que o nodo S quisesse se comunicar com o nodo X, rotas apropriadas seriam S – A – B – D – E – F – X ou S – A – C – D – E – F – X. Porém, se um nodo I enviasse pacotes RREP forjados para D, se passando por C, ou seja, alterando o endereço do pacote para o endereço de C, e informando que possui uma rota para X menos custosa do que a rota oferecida pelo nodo E, um *loop* se formaria.

Dessa forma, toda informação enviada por S, destinada a X, passaria pelos nós A – B – D – C, mas nunca alcançaria seu destino final, retornando sempre de D para C.

Mensagens de erro forjadas

Os protocolos de roteamento implementam técnicas para manter atualizadas as informações sobre as rotas que possuem em suas tabelas de roteamento [Bantz, 1994].

Essas técnicas têm como objetivo perceber o mais rapidamente possível mudanças na topologia da rede. Usando a figura 17, como exemplo, se o nodo C se mover para fora do alcance da transmissão de algum de seus vizinhos, como por exemplo, B, ele conseqüentemente deixaria de ser vizinho de B. Então B teria que perceber essa mudança e atualizar sua tabela, mantendo a rota caso ela ainda estivesse sendo usada ou removendo-a caso ela não fosse mais necessária. Se uma rota alternativa não for achada, uma mensagem de erro deverá ser gerada por B, informando que o link entre ele e D não existe mais. Essa mensagem deveria ser propagada para os nós anteriores a B, nesse caso o nodo A.

Um ataque poderia ser executado forjando mensagens de erro indicando que um determinado link teria se desfeito. Imaginemos que o link entre B e C nunca tivesse se desfeito. Um atacante, I, poderia enviar mensagens de erro para A, usando o endereço do nodo B, indicando a quebra do link entre B e C que, na realidade, nunca teria acontecido. I poderia fazer isso continuamente, impedindo a comunicação entre os nós, realizando, assim, um ataque de negação de serviço.

Melhorando As Tabelas De Rotas

Alguns protocolos, como por exemplo, o Dynamic Source Routing, tentam apreender novas rotas, observando os pacotes que trafegam na rede [Bantz, 1994].

Quando recebe, de maneira promíscua, um pacote que contém informações sobre uma determinada rota, que não existe em sua tabela de roteamento, um nodo, executando esse tipo de protocolo, é capaz de incluir essa nova rota em sua tabela.

Essa técnica garante um ganho na eficiência do protocolo de roteamento, porém abre mais uma brecha na segurança. Um atacante poderia se aproveitar dessa característica do protocolo para forjar mensagens com rotas falsas, utilizando endereços também forjados. Quando os nós da rede recebessem esses pacotes e tentassem aprender com eles, estariam, na verdade, "envenenando" suas tabelas de roteamento com rotas falsificadas.

Essa característica do protocolo é opcional, podendo ser desativada sem que o protocolo deixe de funcionar. Mas, a conseqüência disso pode ser uma perda considerável em sua eficiência.

Ataques de fabricação

O ataque de fabricação também é chamado de geração de falsas mensagens de roteamento. Tais ataques tornam muito difícil a verificação da validade das mensagens, especialmente no caso de mensagens de erro fabricadas, que alegam que um vizinho não pode ser contatado.

4.1.7. Ataques gerais

Os ataques, para diversos autores [Hu, 2002b], [Kumar, 2001], [Bradley, 1997], [Wang, 2003] a protocolos de roteamento de redes *Ad Hoc* geralmente podem ser classificados em uma das duas categorias a seguir [Hu, 2002b], [Beard, 2002]: ataques de interrupção de roteamento e ataques de consumo de recursos. Em um ataque de interrupção de roteamento, o invasor tenta fazer com que pacotes de dados legítimos sejam roteados da maneiras errada. Em um ataque de consumo de recursos, o invasor injeta pacotes na rede em uma tentativa de consumir recursos da rede (como largura de banda), ou de consumir recursos dos nós, tais como memória (armazenamento) ou poder computacional.

4.1.8. Ataque Black hole

Da mesma forma, um invasor pode criar um "buraco negro" de roteamento e excluir todos os pacotes através desse buraco. Enviando pacotes de roteamento falsificados, o invasor pode rotear para si mesmo todos os pacotes com um destino qualquer e, então, descartá-los; ou, o invasor pode fazer com que a rota em todos os nós, em uma área da rede, aponte para "dentro" daquela área, quando, na verdade, o destino é fora dela. Como um caso especial de buraco negro (black hole), um invasor poderia criar um "buraco cinza" (gray hole), através do qual ele excluiria seletivamente alguns pacotes (mas não todos), encaminhando, por exemplo, apenas os pacotes de roteamento, enquanto excluiria os pacotes de dados.

4.1.9. Detours (desvios)

Um invasor pode também tentar fazer com que um nó use *detours* (rotas sub-ótimas) ou, tentar dividir a rede injetando pacotes de roteamento falsificados para evitar que um conjunto de nós alcance outro. Um invasor pode tentar fazer uma rota, através de si mesma, parecer mais longa, adicionando nós virtuais a essa rota, já que uma rota mais curta existe e que, de outra forma, teria sido utilizada. Em protocolos de roteamento de redes *Ad Hoc,* que buscam manter o rastreamento de nós maliciosos em uma "lista negra" (*blacklist*) em cada nó, um invasor pode

extorquir um nó bom, fazendo com que outros nós bons adicionem aquele nó as suas listas negras, evitando-o em suas rotas.

4.1.10. Exclusão de pacotes

Um invasor pode tentar reduzir a quantia de informação de roteamento disponível para outros nós, causando defeitos no anúncio de certas rotas, destruindo ou descartando pacotes de roteamento ou partes desses pacotes. Normalmente, não existe defesa contra esse tipo de ataque, já que o invasor poderia também, nesse caso, excluir pacotes de dados enviados para esses destinos.

4.1.11. Ataque Replay (por reprodução)

Um invasor pode montar um ataque replay enviando um anúncio antigo para algum nó, em uma tentativa de fazer esse nó atualizar sua tabela de roteamento com rotas velhas.

4.1.12. Ataque Invisível

Um invasor pode tentar fazer a rota, através dela mesma, parecer mais curta, ao tornar-se invisível para a rota. Esse ataque possui efeitos limitados na rede se o invasor estiver sempre invisível. A topologia e a informação de roteamento da rede não são confidenciais no ataque.

4.1.13. Ataque de Consumo de Recurso

Um invasor pode injetar dados ou pacotes de controle extra na rede, para que isso consuma largura de banda ou recursos quando outros nós processarem e remeterem pacotes, especialmente sobre *detours* ou *loops* de roteamento.

4.1.14. Vertex cut (vértice de corte)

Nesse ataque, a rede é dividida em duas. O invasor pode tentar extrair o máximo de recurso dos nós em ambos os lados do vértice de corte, remetendo, por exemplo, apenas os pacotes de roteamento e não os pacotes de dados, de forma que os nós desperdicem energia remetendo pacotes para o vértice de corte, onde eles serão excluídos.

4.1.15. Ataque Rushing

O ataque *rushing* é um ataque malicioso direcionado contra protocolos de roteamento *on-demand* que utilizam supressão de duplicatas em cada nó [Hu and Perring and Johnson, 2002]. Um invasor espalha pedidos de rota (ROUTE REQUESTs - RREQ) rapidamente por toda a rede, suprimindo qualquer pedido de rota legítimo, posterior ao espalhado, uma vez que os nós excluem os pedidos duplicados, devido à ordem de supressão de duplicatas.

4.1.16. Denial of Service

Ataques de Denail of Service (D.o.S – Negativa de Serviço), como o próprio nome indica, procuram tornar algum recurso ou serviço indisponível. Em redes sem fio, estes ataques podem ser tanto mais perturbadores quanto maior for sua sofisticação. Estes ataques podem ser disparados de qualquer lugar dentro da área de cobertura de rede *Wireless*. Por exemplo, um atacante pode se passar por outro, com o mesmo SSID e endereço MAC de um nodo válido, e inundar a rede com pedidos de dissociação. Estes pedidos fazem com que os clientes sejam obrigados a se desassociarem e se re-associarem. Enviando as requisições de dissociação, em períodos curtos de tempo, o D.o.S é concretizado.

4.1.17. Wardriving

O termo *wardriving* foi escolhido por Peter Shipley (http://www.dis.org/shipley/) para nomear a atividade de dirigir um automóvel à procura de redes sem fio abertas, passíveis de invasão. Para efetuar a prática do *wardriving*, são necessários: um automóvel, um computador, uma placa Ethernet configurada no modo "promíscuo" (o dispositivo efetua a interceptação e leitura dos pacotes de comunicação de maneira completa), e um tipo de antena, que pode ser

posicionada dentro ou fora do veículo (uma lata de batatas fritas costuma ser utilizada para a construção de antenas). Tal atividade não é danosa em si, pois alguns se contentam em encontrar a rede *Wireless* desprotegida, enquanto outros efetuam *login* e uso destas redes, o que já ultrapassa o escopo da atividade. No Brasil, já foi constada a desproteção de uma rede *Wireless* pertencente a um banco internacional, na zona Sul de São Paulo, mediante *wardriving*, entre outros casos semelhantes.

4.1.18. Warchalking

Inspirado na prática surgida na Grande Depressão norte-americana, quando andarilhos desempregados (conhecidos como "hobos") criaram uma linguagem de marcas de giz ou carvão em cercas, calçadas e paredes, indicando uns aos outros o que esperar de determinados lugares, casas ou instituições, por exemplo, onde poderiam conseguir comida e abrigo temporário, o *warchalking* é a prática de escrever símbolos, indicando a existência de redes *Wireless* e informando sobre suas configurações. As marcas, usualmente feitas em giz em calçadas, indicam a posição de redes sem fio, facilitando a localização para uso de conexões alheias pelos simpatizantes da idéia.

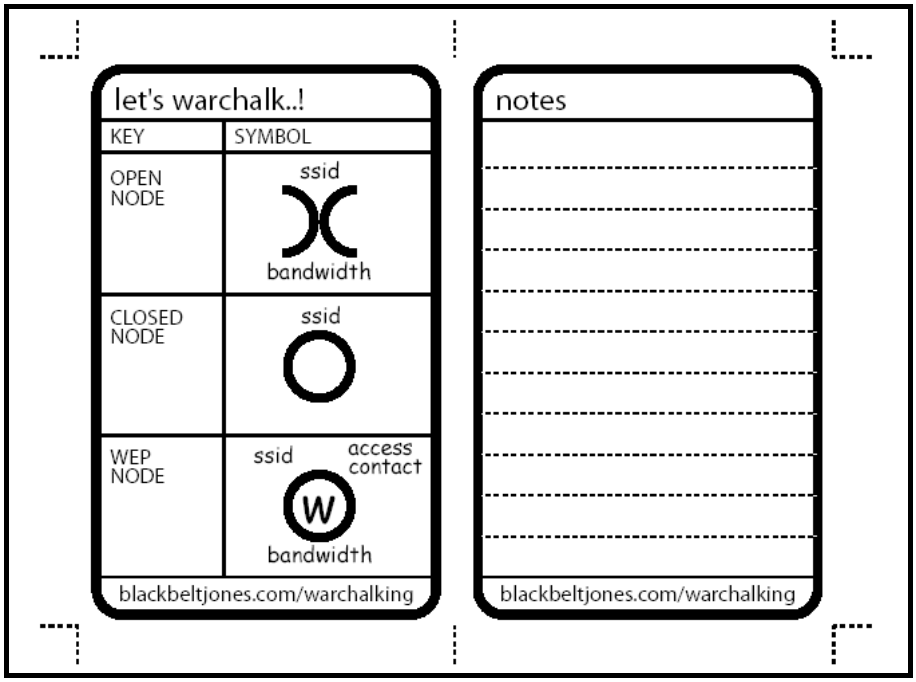


FIGURA 18 – SIMBOLOGIA UTILIZADA NA PRÁTICA DO *WARCHALKING*.
Fonte: *Warchalking - Collaborativelly creating a hobo-language for free Wireless networking,* [Wamis, 2004].

Nas legendas da figura 18, que é uma etiqueta adesiva, o *Open Node* significa que a rede é vulnerável, *Closed Node* serve para uma rede fechada, e a letra W, dentro do círculo, informa que a rede *Wireless* utiliza o padrão de segurança WEP (*Wireless Equivalent Privacy*), com presença de criptografia.

Em cima de cada símbolo, temos o SSID (*Service Set Identifier*), que funciona como uma senha para o *login* na rede, obtidos através de softwares próprios, conhecidos como *sniffers*.

Aqui, novamente, os praticantes do *warchalking* alegam que não pode ser configurado crime rabiscar em calçadas e paredes, e não incitam o uso danoso de redes *Wireless* alheias.

4.1.19. Wormholes

Em um ataque *wormhole* [Hu, 2003], um invasor recebe pacotes em um ponto na rede. A seguir, propaga esses pacotes para um outro ponto na rede, através de um túnel e, então, os reproduz na rede a partir daquele ponto. Em túneis com distâncias mais longas do que o intervalo normal de transmissão *Wireless* de um único salto, é simples para o invasor fazer com que o pacote enviado pelo túnel chegue mais cedo do que os outros pacotes transmitidos por uma rota de *multihops* normal (por exemplo, através da utilização de um único *link Wireless* direcional de longo alcance ou através de um *link* cabeado direto para um invasor em conluio). Também é possível que o invasor remeta cada bit diretamente pelo *wormhole* (o túnel criado pelo invasor), sem esperar que um pacote inteiro seja recebido antes de iniciar o envio de bits do pacote através do túnel, de modo a minimizar o atraso introduzido pelo *wormhole*.

Se o invasor realiza essas conexões por túneis com objetivos honestos e confiáveis, não haverá problemas e nenhum dano é causado. O invasor, na verdade, realiza um serviço muito útil de conexão mais eficiente na rede. Entretanto, o *wormhole* coloca o invasor em uma posição muito poderosa, em relação aos outros nós na rede, sendo que o invasor pode explorar essa posição em uma grande variedade de formas. O invasor pode ainda realizar o ataque, mesmo em casos em que a comunicação na rede oferece confidencialidade e autenticidade, e mesmo que não possua nenhuma chave criptográfica.

O ataque *wormhole* é particularmente perigoso para muitos protocolos de roteamento em redes *Ad Hoc* nos quais os nós que escutam a transmissão de um pacote, diretamente a partir de algum nó, consideram-se dentro do alcance (e também um vizinho) daquele nó. Por exemplo, quando utilizado contra um protocolo de roteamento reativo (*on-demand)* como o DSR ou AODV, uma poderosa aplicação do ataque *wormhole* pode ser conseguida quando cada pacote de pedido de rota (ROUTE REQUEST) é enviado por túnel diretamente para o nó destino, que é alvo do pedido (REQUEST). Quando os vizinhos do nó destino escutam esse pacote de pedido (REQUEST), eles seguem o processamento normal do protocolo de roteamento para retransmitir, por broadcast, aquela cópia do pedido (REQUEST) e, então, descartam, sem processar, todos os outros pacotes de pedido de rota (ROUTE REQUEST), recebidos e originados a partir desse mesmo descobrimento de rota. Dessa maneira, esse ataque evita que seja descoberta qualquer outra rota que não seja através do wormhole e, se o ataque for próximo ao iniciador do Descobrimento de Rota, o invasor pode até mesmo evitar que rotas com mais de dois hops de comprimento sejam descobertas. Formas possíveis de exploração do *wormhole* pelo invasor incluem o descarte, ao invés de propagação, de todos os pacotes de dados, a criação de um ataque permanente de Denial-of-Service (DoS – negação de serviço) - onde nenhuma outra rota para o destino pode ser descoberta enquanto o invasor mantiver o *wormhole* para pacotes de pedido de rota (ROUTE REQUEST) - ou um descarte seletivo ou modificação de determinados pacotes de dados.

A funcionalidade de descobrimento de vizinho dos protocolos de roteamento periódicos (pró-ativos) - como o DSDV, OLSR e TBRPF, dependem da recepção de pacotes por *broadcast,* como um mecanismo para a detecção de vizinho, e são também extremamente vulneráveis a esse ataque. Por exemplo, o OLSR e o TBRPF usam pacotes *HELLO* para a detecção de vizinho, de forma que se um invasor envia por túnel, para B, todos os pacotes *HELLO* transmitidos por A, e envia por túnel, para A, todos os pacotes *HELLO* transmitidos por B. Então A e B irão acreditar que são vizinhos, o que levará os protocolos de roteamento a falharem no descobrimento de rotas.

Para o DSDV, se cada anúncio de roteamento remetido pelo nó A fosse enviado por túnel para o nó B, e vice versa, então A e B acreditariam serem vizinhos. Se eles não estiverem dentro do alcance da transmissão *Wireless*, eles não serão capazes de se comunicar. Além disso, se a melhor rota existente de A para B tiver pelo menos 2n + 2 hops de comprimento, então qualquer nó dentro do limite de n *hops* de A seria incapaz de se comunicar com B, e qualquer nó dentro do limite de n *hops* de B seria incapaz de se comunicar com A. De outra forma,

suponhamos que C estivesse dentro do limite de n *hops* de A, mas tivesse uma rota válida para B. Uma vez que A anuncia uma métrica 1 rota para B, C escutaria uma métrica n + 1 rota para B. C tomará essa rota se não estiver dentro do limite de n + 1 hops de B; nesse caso haveria um caminho n-hop de A para C, e um caminho n + 1-hop de C para B, contradizendo a premissa de que o melhor caminho real de A para B tem pelo menos 2n + 2 hops de comprimento.

Detectando Ataques Wormhole

Na pesquisa realizada por Hu [Hu, 2003], é apresentada uma solução para detecção de ataques *wormhole*, introduzido à noção de um *packet leash* como um mecanismo geral para detectar e, conseqüentemente, combater os ataques *wormhole*. Um *leash* é qualquer informação que é adicionada a um pacote com o objetivo de restringir a distância de transmissão máxima permitida do pacote. Fazemos a distinção entre *leashes* geográficos e temporais. Um *leash* geográfico garante que o destinatário do pacote esteja dentro de certa distância do remetente. Um *leash* temporal garante que o pacote tenha um limite superior em seu tempo de vida, o que restringe a distância máxima de viagem, uma vez que o pacote pode viajar no máximo à velocidade da luz. Qualquer tipo de *leash* pode impedir o ataque *wormhole* por possibilitar que o destinatário de um pacote detecte se o pacote viajou além do que o *leash* permite.

Leashes Geográficos

Para construir um leash geográfico, em geral, cada nó precisa conhecer sua própria localização e todos os nós precisam ter relógios fracamente sincronizados. Quando um pacote é enviado, o nó remetente (sender) inclui no pacote sua própria localização, ps, e o tempo em que ele enviou o pacote, ts; quando um pacote é recebido, o nó receptor compara esses valores com a sua própria localização, pr, e o tempo em que recebeu o pacote, tr. Se os relógios do remetente e do receptor (do *sender* e do *receiver*) estão sincronizados em $\pm \Delta$, e ν é um limite superior na velocidade de qualquer nó, então o receptor pode computar um limite superior na distância entre ele e o remetente, dsr. Mais especificamente, baseado no timestamp ts no pacote, no tempo local de recepção (*local receive time*) tr, no máximo erro relativo na informação de localização δ, e nas localizações do receptor pr e do remetente ps, então dsr pode ser limitado por $dsr \leq \| ps - pr \| + 2\nu \cdot (tr - ts + \Delta) + \delta$. Um esquema de assinatura digital regular, ou outra técnica de

autenticação, pode ser usada para permitir ao receptor autenticar a localização e o *timestamp* no pacote recebido [Hu, 2003].

Leashes Temporais

Para construir um leash temporal, em geral, todos os nós precisam ter relógios rigorosamente sincronizados, de forma que a diferença máxima entre os relógios de dois nós quaisquer seja Δ. O valor do parâmetro Δ deve ser conhecido por todos os nós na rede, e, para leashes temporais, geralmente deve ser da ordem de alguns micro segundos ou mesmo de centenas de nano segundos. Esse nível de sincronização de tempo pode ser atingido, hoje, com hardware de prateleira (off-the-shelf hardware). Apesar de tal hardware não ser, atualmente, uma parte comum dos nós de rede *Ad Hoc*, ele pode ser desdobrado em redes *Ad Hoc* atuais e, espera-se que seja mais amplamente utilizado em sistemas futuros, como redução de despesas, tamanho, peso, e consumo de energia. Além disso, o próprio sinal de sincronização de tempo, em tais sistemas, pode estar sujeito a certos ataques. Hardwares esotéricos, como *cesium-beam clocks*, *rubidium clocks*, e *hydrogen maser clocks,* também poderiam ser utilizados em aplicações especiais atuais, para possibilitar sincronização de tempo suficientemente preciso por meses. Apesar de nossa necessidade geral de sincronização de tempo ser, na verdade, uma restrição na aplicabilidade dos *leashes* temporais, para aplicações que requerem defesa contra o ataque *wormhole*, essa necessidade é justificada pela seriedade do ataque e sua potencial interrupção no pretendido funcionamento da rede [Hu, 2003].

Para utilizar *leashes* temporais, quando um pacote é enviado, o nó remetente inclui no pacote o tempo em que ele enviou o pacote, ts; quando o pacote é recebido, o nó receptor compara esse valor com o tempo em que ele recebeu o pacote, tr. O receptor está apto para detectar se o pacote viajou muito longe, baseado no tempo de transmissão declarado e na velocidade da luz. Alternativamente, um *leash* temporal pode ser construído, ao contrário do que foi descrito, incluindo-se no pacote um tempo de expiração, após o qual o receptor não deverá aceitar o pacote; baseado na máxima distância de transmissão permitida e na velocidade da luz, o remetente fixa esse tempo de expiração no pacote, como um offset, a partir do tempo em que ele envia o pacote. Da mesma forma que o uso de um *leash* geográfico, um esquema de assinatura digital regular ou outra técnica de autenticação pode ser usado para permitir que o receptor autentique um *timestamp* ou tempo de expiração no pacote recebido [Hu, 2003].

5. Ambiente de Simulação

Nesta seção, serão descritas as ferramentas de simulação e de análise, algumas características de desempenho e a metodologia utilizada para realizar a análise comparativa de desempenho dos protocolos de roteamento estudados.

5.1. Metodologia

A utilização de simulações é uma das técnicas mais difundidas para avaliação de desempenho de sistemas em geral, representando um importante papel em várias áreas de conhecimento. Não há grandes projetos sem uma série de simulações de seu comportamento, avaliação das respostas do sistema para as mais diversas condições de exposição das simulações.

O objetivo principal de realizar simulações é modelar, computacionalmente, um sistema do mundo real e, então, avaliar as simulações que acontecem no ambiente real. Diante disso, podemos dizer que as simulações são o melhor caminho para se obter boas informações do comportamento do projeto a ser implementado.

A simulação é também um grande aliado do custo/beneficio de um projeto, levando em consideração que o custo de simulação, normalmente, representa um valor muito baixo, se comparado com a execução do projeto, além de reduzir as chances de erros na execução e na própria elaboração do projeto.

Para identificar os impactos da incorporação de segurança nos protolos de roteamento de redes, são realizadas simulações para coletar dados das métricas de: vazão, latência, *jitter*, descarte de pacotes. Através dos resultados coletados das simulações, será possível obter informações de desempenho das redes *Ad Hoc*. A técnica de simulação é utilizada com grande freqüência pela flexibilidade em testar cenários variados, incluindo o comportamento de protocolos, novas tecnologias e efeito de diferentes topologias [Silva, 2004].

5.2. Tipos De Simulação

Um item importante a ser considerado, antes de ser dar início a uma simulação, é o tipo de método de simulação mais adequado ao problema. Na literatura, diversos tipos de simulações podem ser encontrados. Porém, para o enfoque das redes de computadores, os métodos mais relevantes compreendem: o método de Monte Carlo, método baseado em traces e o método fundamentado em eventos discretos. Dentre os três tipos de simulações, o método baseado em traces é o que contempla as exigências dos modelos que serão simulados.

5.2.1. Simulação De Monte Carlo

O método de Monte Carlo é baseado na simulação de variáveis aleatórias para resolução de problemas. Este é um tipo de simulação que serve para modelar fenômenos probabilísticos invariantes no tempo. O comportamento de um sistema pode ser caracterizado por distribuições de probabilidades e seus parâmetros. No entanto, na maioria das vezes, os parâmetros são desconhecidos, e simulações de Monte Carlo podem ser aplicadas na obtenção de estimativas, através da realização de várias replicações de um experimento. Por exemplo, suponha-se que o tamanho dos pacotes gerados por uma determinada aplicação segue uma distribuição qualquer fX(x). Para determinar o tamanho médio dos pacotes, pode-se utilizar uma simulação de Monte Carlo para gerar várias ocorrências da variável X e, depois, calcular a média amostral que serve de estimativa para a média E(X) [Silva, 2004].

5.2.2. Simulação Discreta Baseada Em Eventos

Uma simulação discreta, baseada em eventos, utiliza um modelo de estados discretos para o sistema. Diferente de um modelo contínuo, no qual os estados que o sistema pode assumir variam continuamente, em um modelo discreto, o sistema só pode assumir um número discreto de valores. É importante notar que, mesmo em uma simulação discreta, o tempo da simulação pode assumir valores discretos ou contínuos. Qualquer simulação discreta, independente de aplicação, deve conter os seguintes componentes: um escalonador de eventos; um mecanismo de

controle do tempo (clock); variáveis globais que descrevem os estados do sistema; rotinas para simular os eventos; rotinas para entrada de parâmetros; rotinas para coletar resultados; rotinas de iniciação; rotinas para gerenciamento dinâmico de memória e um programa principal [Silva, 2004].

5.2.3. Simulação Baseada Em Traces

Uma simulação baseada em traces tem como entrada um registro que contém eventos, ordenados no tempo, observados em um sistema real. Esses registros são chamados de traces. Este tipo de simulação é geralmente utilizado na análise de algoritmos de alocação de recursos. Neste caso, um trace, contendo a demanda por um determinado recurso, é utilizado como entrada da simulação, a qual pode incluir diferentes algoritmos para serem avaliados sob as mesmas condições de demanda. Uma característica importante é a credibilidade. Um trace, contendo os acessos feitos a um determinado serviço na Internet, tem uma maior credibilidade do que informações geradas randomicamente através de alguma distribuição. Um dos principais problemas dos traces é o tamanho. Os traces são, geralmente, seqüências longas e exigem um considerável tempo computacional para serem processados. Outro problema é a dificuldade de variação da carga de trabalho aplicada [Silva, 2004].

5.3. Características de Desempenho

Existem várias métricas quantitativas que devem ser analisadas para avaliar o desempenho de um protocolo de roteamento. A RFC-2501[Corson, 1999] define quatro métricas que são consideradas básicas, que podem ser utilizadas para avaliar qualquer protocolo de roteamento. As métricas definidas na RFC-2501[Corson, 1999] são:

Retardo (Delay end-to-end): um dos principais parâmetros de desempenho da rede é o atraso. Denota o tempo decorrido para transmitir um bloco de dados de um nó fonte até que ele alcance o nó destino. Na camada de aplicação, o retardo é a diferença de tempo (fim-a-fim) transcorrida entre a geração do dado no transmissor e a sua apresentação no receptor. Essa

diferença inclui parcelas referentes ao processamento nos nós intermediários (roteadores, comutadores) e finais (*endpoints*), a disputa pelo acesso ao meio nos enlaces compartilhados e ao tempo de propagação no meio físico. Caso a conexão entre transmissor e receptor envolva múltiplos saltos, como é comum em redes comutadas por pacotes, a soma de todos os retardos, salto-a-salto, mais o retardo de processamento, deve ser igual ou inferior ao retardo fim-a-fim desejado. Normalmente deseja-se limitar algum parâmetro relativo à curva de distribuição do retardo, como um valor médio, máximo, ou um percentual. Nessa métrica, estão incluídos o tempo de aquisição da rota (protocolos reativos) e o tempo de transmissão sobre o meio *Wireless*. O tempo de computação (processamento) não é levado em consideração. Quanto menor a perda de pacotes (*dropped packets*), maior a eficiência da rede.

Vazão (throughput): a vazão em uma rede é a largura de banda efetiva ou a taxa efetiva de bits por segundo, ou seja, a quantidade de dados transmitidos com sucesso por unidade de tempo. Pode ser definida como sendo a diferença entre a taxa de bits de ligação e os vários *overheads* (sobrecarga). A vazão, na maioria das redes, sofre variações no decorrer do tempo. Em algumas situações, a vazão pode se alterar rapidamente, devido às falhas nos nós da rede ou linhas, ou devido ao congestionamento, quando grandes fluxos de dados são introduzidos na rede. Pacotes excluídos, bem como pacotes de roteamento e de sinalização, não são levados em consideração. Essa métrica reflete a efetividade do protocolo.

Pacotes excluídos: Pacotes podem ser descartados em seu caminho para o destino devido a diferentes fatores, tais como a não existência de rota para o destino ou a interrupção do período de aquisição de rota. Essa métrica é a relação entre a quantia de pacotes excluídos e a quantia de pacotes transmitidos.

Overhead: Essa métrica reflete a eficiência do protocolo. Uma vez que o protocolo de roteamento atingiu sua meta que é proporcionar rotas para entrega de pacotes de dados, temos que medir como o protocolo o fez. Os pacotes de controle irão competir com pacotes de dados por acesso ao meio, de forma que ele possa também interferir em outras métricas. Conseqüentemente, o protocolo de roteamento, que envia a mínima quantia possível de pacotes de controle, é o mais eficiente. Essa métrica é representada pela quantia bruta de pacotes de roteamento transmitidos.

1. Throughput de Dados End-to-end e delay

Esta métrica envolve analisar a eficiência do roteamento dos dados. As medidas estatísticas (por exemplo, médias, variações e distribuições) são essenciais. Estas são usadas para analisar a eficácia de uma política do roteamento como medida da perspectiva externa (a perspectiva de outras políticas que utilizam o roteamento).

2. Tempo de aquisição de rota

Ele mede o tempo necessário para estabelecer a rota. É uma medida fim-a-fim. O tempo necessário para estabelecer uma rota é especialmente interessante para os protocolos de roteamento reativos (on-demand).

3. Percentual de entregas fora-de-ordem.

Ele mede o desempenho do roteamento sem conexão. É uma métrica interessante, especialmente do ponto de vista da camada do transporte (por exemplo, TCP), desde que a camada de transporte prefere, na maior parte, a entrega em-ordem.

4. Eficiência

Refere-se ao desempenho da política interna de roteamento. Se o controle e o tráfego de dados usam o mesmo canal de transmissão, o excessivo controle do tráfego provavelmente afetará a eficiência de uma política interna. Quando analisamos a performance de um protoloco de roteamento *Ad Hoc*, alguns parâmetros do contexto da rede serão cuidadosamente considerados. A RFC-2501[Corson, 1999] define os seguintes:

Tamanho da rede: Significa mensurar o número de nós em uma rede

Conectividade da rede: Refere-se ao número médio de nós vizinhos.

Taxa de mudança topológica: Taxa de mudança de topologia da rede.

Capacidade do Link: Refere-sea velocidade efetiva da rede. É medida em bits por segundo.

Fração dos links unidirecionais: Indica como o número de links unidirecional presentes na rede afeta o desempenho do protocolo.

Tráfego padrão: Indica como um protocolo pode se adaptar aos testes padrões de tráfego não-uniformes ou estouros.

Mobilidade: Trata de avaliar a correlação da topologia temporal ou espacial, relaevante no desempenho do protocolo de roteamento.

Fração e freqüência de nós em *sleeping*: Verifica a eficiência de um protocolo no controle daa situação, quando os nós presentes nas redes encontram-se em estado sleeping.

5.4. O Simulador (Network Simulator - NS-2)

Para a realização das simulações, adotou-se o simulador *Network Simulator* (NS) [Network Simulator, 2004], versão 2.27. O NS foi concebido em 1989, a partir de uma variação do *REAL Network Simulator*, um projeto da Cornell University, EUA. Atualmente, o desenvolvimento do NS é suportado pelo DARPA (Defense Advanced Research Projects Agency, EUA), através do projeto SAMAN, e pela NSF (National Science Foundation, EUA), através do projeto CONSER, em colaboração com outros pesquisadores como o centro ICIR. O simulador já recebeu apoio do Lawrence Berkeley National Laboratory, do Xerox PARC (Palo Alto Research Center), da Universidade da Califórnia em Berkeley, Sun Microsystems e também agrega diversos módulos contruídos por pesquisadores independentes [Portnoi e Araújo, 2002]. É um software de código livre, fornecido gratuitamente [Network Simulator, 2004].

O NS é um simulador de eventos discretos, focado para o desenvolvimento de pesquisas em redes de computadores. Ele prevê suporte a TCP e variantes do protocolo, *multicast*, redes sem fio (*Wireless*), roteamento e satélite. Implementa filas de roteamento tipo *droptail*, Diffserv RE, *fair queueing* (FQ), *stochastic fair queueing* (SFQ), *class-based queueing* (CBQ), dentre outras. Tem facilidades de tracing, que é a coleta e registro de dados de cada evento da simulação, para análise posterior. Possui um visualizador gráfico para animações da simulação (nam – network animator), timers e escalonadores, modelos para controle de erros e algumas ferramentas matemáticas como gerador de números aleatórios e integrais para cálculos estatísticos. Inclui também uma ferramenta de plotagem, o xgraph, e vários tipos de geradores de tráfego [Network Simulator, 2004].

Existem versões do simulador para as plataformas Windows, utilizando o *Linux Virtual Machine* (Cygwin) e Unix (FreeBSD, SunOS, Solaris, Linux, dentre outras).

O NS foi desenvolvido na linguagem orientada a objetos C++, de forma modular. O uso desta linguagem nos módulos confere velocidade e mais praticidade na implementação de protocolos e modificação de classes. A interface com o usuário, configuração, estabelecimento de parâmetros e manipulação de objetos e classes é feita em modo texto, através da linguagem interpretada Otcl, que também é orientada a objetos.

O simulador de rede NS é um simulador de eventos discretos na rede e é capaz de simular diferentes tipos de redes. A camada física simulada é transmissão de rádio; o NS simula a propagação de controles de rádio (radio controls). O uso de valores de energia de transmissão, *thresholds* e banda utilizada foram escolhidos de acordo com a interface *Wireless* do manufaturador Lucent, modelo WaveLAN, dando aos nós um alcance de aproximadamente 250 metros no modelo de atenuação e interferência utilizado *Two-Ray Ground*, que simula um ambiente outdoor e a interface elevada a 1,5m do chão.

O esquema de acesso ao meio é DCF (Distribuído Coordenado Ponto) com CSMA/CA (Acesso Múltiplo com Verificação de Portadora/Colisão Permitida), também do padrão IEEE 802.11.

O arquivo de saída de uma simulação no NS-2 [Network Simulator, 2004] é um trace file sustentando uma linha para cada evento que ocorreu durante a simulação. Os eventos possíveis são transmissão, recepção, descarte e envio de um pacote. Os eventos são registrados para todas as camadas na pilha de cada nó. Cada pacote gerado para um dado nó carrega uma identificação única, o que permite ao pacote ser seguido através dos trace files. Isso permite que meçamos, por exemplo, o delay (atraso) do pacote como sendo o tempo que ele leva para atingir seu destino.

O ambiente simulado no NS-2 é simplificado. Ele não possui obstáculos físicos para nós e/ou controles de rádio (radio controls), nem nenhum tipo de interferência externa para os controles de rádio (radio controls). Com isso, pode-se presumir que o meio de transmissão é mais confiável na simulação do que no mundo real e que todos os links são bi-direcionais.

Estrutura do Network Simulator – NS

A interface entre o usuário e o NS dá-se através da linguagem script OTcl. Segundo os desenvolvedores, a divisão em duas linguagens (OTcl e C++) objetiva dar ao simulador tanto velocidade e poder, quanto flexibilidade e facilidade de mudança de parâmetros. A linguagem C++, na qual o simulador foi escrito, confere-lhe velocidade, mas esta torna-se lenta para manipulação constante ou mudança de parâmetros. OTcl, por ser interpretada, é bem mais lenta, porém pode ser facilmente alterada. Além do mais, os objetos compilados são disponibilizados para o interpretador OTcl por linkagem, o que virtualmente cria um objeto OTcl para cada objeto C++, que podem ser manipulados através das facilidades da Otcl. Um usuário comum atua no perímetro "tcl", escrevendo scripts em OTcl e executando simulações. Os escalonadores de eventos e os componentes de rede são implementados em C++ e disponibilizados ao interpretador OTcl através de uma replicação feita pela camada tclcl, que recria os objetos C++ em objetos OTcl, e que podem, finalmente, ser manipulados por esta última (processo denominado linkage). Todo o conjunto constitui-se no NS, que é um interpretador de OTcl com bibliotecas de simulação para redes de computadores [Network Simulator, 2004].

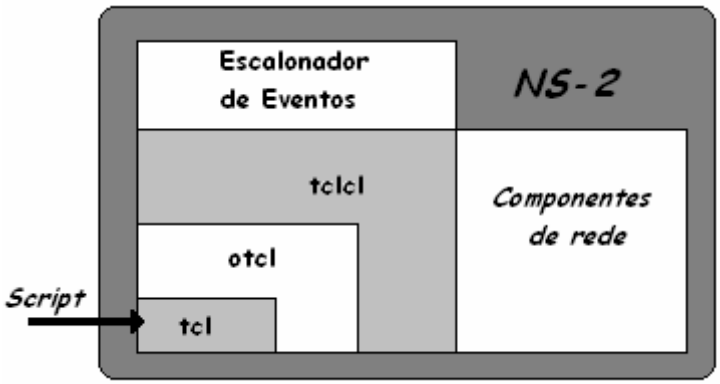


FIGURA 19 – ARQUITETURA DO NS.

5.5. A Ferramenta de Análise

A análise dos resultados das simulações foi realizada com a ferramenta de análise *Trace Graph*, a qual lê e interpreta os arquivos gerados pelo simulador. Este software foi desenvolvido em Wroclaw University of Technology – na Polônia, pelo estudante Jaroslaw Malek. A ferramenta encontra-se, atualmente, na versão 2.02 [Malek, 2005].

Este software permite que sejam efetuadas as análises comparativas de todas as métricas sugeridas na RFC 2501 [Corson, 1999] para avaliação de desempenho de um protocolo de rede, que fazem parte do escopo deste trabalho, sendo elas:

- **Vazão (throughput):** a vazão em uma rede é a largura de banda efetiva ou a taxa de bits efetiva, ou seja, a quantidade de dados transmitidos com sucesso por unidade de tempo.

- **Latência**: é o tempo que um pacote IP leva para ir e voltar (*round-trip*) de um ponto a outro da rede. É normalmente medida em milissegundos. Quanto menor a latência, melhor o tempo de resposta da rede.

- **Variação de Retardo (Jitter)**: definido como sendo a diferença entre o atraso do pacote atual e o do próximo pacote. Sendo S(i) o tempo em que o pacote i foi enviado e R(i) o tempo em que o pacote i foi recebido, o jitter (i) =|(R(i+1)-S(i+1))-(R(i)-S(i))|.

- **Taxa de Erros ou Perdas de Pacotes**: definido como a taxa de sucesso na transmissão de pacotes IP entre dois pontos da rede. É normalmente exibida como uma porcentagem, indicando o percentual de pacotes (ou bytes) perdidos durante a simulação. Quanto menor a perda de pacote (*dropped packets*), maior a eficiência da rede.

6. Simulações e Resultados

Neste capítulo, estão descritos detalhes da implementação dos *scripts*; os modelos simulados foram analisados para registrar o desempenho dos resultados obtidos.

6.1. Contexto das Simulações

O objetivo deste trabalho consistiu em analisar os impactos da implementação da segurança nos protocolos de roteamente de redes *Wireless Ad Hoc*.

A verificação estatística de cada um dos fatores de desempenho dos protocolos de roteamento representou análise importante para checar, comparativamente, a evolução dos algoritmos de roteamento após a agregação de segurança. Para este fim, foram escolhidos protocolos de roteamento onde a sua evolução partiu de protocolos não seguros.

Na análise de desempenho realizou-se uma verificação estatística de cada fator que influencia na performance de um protocolo. As métricas consideradas foram: vazão, latência, variação de retardo (*jitter*) e descarte de pacotes. Os resultados obtidos nos protocolos seguros foram confrontados com os resultados obtidos quando obtidos nos protocolos não seguros.

Para realizar as simulações buscou-se criar um cenário muito próximo do mundo real, onde o ambiente criado atendia às necessidades do usuário e também dos dispositivos *Wireless,* envolvidos nas simulações.

Os padrões de movimentação e cenários de comunicação utilizados nos experimentos foram gerados através dos *scripts cbrgen.tcl* e *setdest,* presentes na distribuição do *Network Simulator 2.27*. Os cenários são utilizados pelo simulador para realizar a comunicação entre os nós da rede. Para cada simulação realizada foram utilizados os mesmos cenários e padrões de movimentos para que as análises de todos os protocolos testados fossem elaboradas sobre uma mesma ótica.

6.2. Modelo de Mobilidade – Random Way-Point

As estações movem-se numa região de acordo com modelo de mobilidade individual de *Way-Point.* O modelo *Random Way-Point* [Yoon and Liu and Noble, 2003] funciona da seguinte forma: antes da simulação, os seguintes parâmetros são definidos pelo usuário: protocolos de roteamento, número de nós, área de simulação, velocidade máxima e tempo de pausa máximo (*maximum pause-time*), tamanho dos pacotes (*packet size*), tempo de simulação. Quando a simulação começa, o modelo define, em um modo aleatório, o destino e a velocidade de cada nó. A velocidade escolhida está necessariamente entre os valores previamente definidos de velocidade máxima e mínima. Quando o nó atinge seu destino, permanece imóvel durante um tempo aleatório, escolhido entre zero e o tempo de pausa máximo definido e, então ele segue em outra direção aleatória, com velocidade também aleatoriamente escolhida.

6.3. Parâmetros de Simulação

Nas realizações das simulações, alguns parâmetros foram variados de acordo com a análise a ser realizada, outros foram fixados. A seguir, são apresentados os parâmetros utilizados nas simulações.

6.3.1. Parâmetros Padrões

Alguns parâmetros podem ser considerados como padrão para as redes *Wireless*. São eles:

Parâmetro	Valor
Tipo de MAC:	802.11
Tipo de camada de ligação	LL
Modelo de antena	*Omni Antenna*
Tipo de canal:	*Wireless Channel*
Tipo de interface de Rede:	*Wireless Phy*
Modelo de propagação:	*Two Ray Ground*

6.3.2. Parâmetros Fixos

Os parâmetros aqui descritos foram fixados para todas as simulações.

- Interface de Fila: Decidiu-se por uma única opção de interface de fila, a *DropTail,* espécie de fila do simulador que implementa o algoritmo de escalonamento FIFO, visto que, conforme apresentado por Horstmann [Horstmann, 2002], não há melhora significativa entre a variação dos esquemas de filas FIFO, SFQ, FQ, DRR.

- Quantidade Máxima de Pacotes na Fila: O número máximo de pacotes na fila foi pré-determinado que seria 10, isto porque quanto maior o número de pacotes que a fila é capaz de suportar, maior será o tempo computado para a latência [Horstmann, 2002].

- Modelo de Mobilidade: o modelo de mobilidade utilizado foi o modelo de mobilidade individual de *Way-Point.*

- Velocidade de Movimentação: Os cenários foram definidos para utilizarem uma velocidade máxima de deslocamento de 15 m/s.

- Topografia: a rede é formada dinamicamente pelas estações contidas nas simulações.

- Dimensões do Ambiente: abrangência máxima de 800m x 800m, o suficiente para simular um ambiente *indoor* de uma empresa com seus vários setores.

- Largura de Banda: A largura de banda utilizada foi de 11Mbps, conforme a padronização do IEEE 802.11b. Este padrão consegue trabalhar com taxas de transmissões de 5 e 11 Mbps.

- Modelo de Tráfego: O modelo de tráfego utilizado nas simulações foi o Dados - CBR (*Constant Bit Rate)*, no qual fez-se uso do protocolo de conexão UDP, sem confirmação.

- Taxa de Transmissão: A taxa de geração de cada fonte é tal que a soma das taxas de todas as fontes pertencentes ao conjunto seja igual à capacidade máxima do

canal. Ou seja, se a capacidade máxima do canal é de 11 Mbps, as N fontes geram uma taxa de 11 Mbps cada uma, totalizando 11 Mbps no canal. Buscou-se saturar o canal com o conjunto de fontes, para verificar a capacidade máxima de transmissão nesse experimento.

- Tempo de Simulação: Para o ambiente proposto, cada execução da simulação teve a duração de 600 segundos, tempo considerado suficiente [Lamia 2003] para avaliação das métricas de desempenho.

- Quantidade de Simulações: Cada execução da simulação foi reaplicada 10 vezes para garantir confiabilidade estatística aos resultados.

6.3.3. Parâmetros Variáveis

Os parâmetros variáveis para cada simulação foram:

- Protocolo de Roteamento: Os protocolos de roteamento escolhidos para realizar as simulações forma o DSR (*Dynamic Source Routing*), Ariadne (*A Secure On-Demand Routing Protocol for Ad Hoc Networks),* DSDV (*Destination-sequenced distance vector),* SEAD (*Secure Efficient Ad Hoc Distance Vector Routing Protocol),* AODV *(Ad Hoc on-demand distance vector)* e o TORA *(Temporally ordered routing algorithm)*. Entretanto, vale salienter que não há um consenso sobre qual é o melhor protocolo de roteamento para redes *Ad Hoc*. Todos possuem suas vantagens e desvantagens de acordo com as características avaliadas.

- Número de Estações Móveis: O número de estações utilizadas foi variado entre 20, 30 e 40 estações (nós).

- Tamanho dos pacotes: O tamanho dos pacotes utilizados foi variado entre 64, 128 e 256 bytes.

Após a especificação destes parâmetros, foram realizadas as simulações de uma rede *Wireless Ad Hoc. O*s resultados obtidos serão descritos nas próximas seções.

6.4. Coleta de Dados

A realização de coleta de dados, necessária para a análise, foi realizada através da implementação de *scripts*. O programa de simulação desenvolvido alimentou um interpretador Otcl e foram gerados arquivos de *traces* que contém os resultados das simulações.

Para o processo de simulação e análise dos dados gerados pelo simulador (*NS*), utilizou-se um computador com processador AMD Athlon ™ 1.2 *GHz*, com 512Mb de memória *RAM*. A análise dos arquiovos de *trace* foram efetuadas na ferramenta *Trace Graph*.

6.5. Realização das análises dos dados coletados

A análise necessária dos dados foi realizada utlizando o software *Trace Graph*. Os arquivos de *trace,* gerados através do *Network Simulator* 2.27, foram processados e os dados estraídos foram levados a uma planilha de cálculo para efetuar os cálculos necessários e graficar os resultados.

6.6. Arquivos para o simulador

Os arquivos de *trace,* gerados através do Simulador 2.27, foram processados e os dados estraídos foram levados a uma planilha de cálculo, a fim de graficar os resultados de forma comparativa. Um dos fatores de maior dificuldade da análise das informações esteve focado nas ferramentas *Trace Graph*, visto que os arquivos de trace são arquivos gerados, que ficam com tamanhos entre 100 a 400 MB, e a ferramenta possuí algumas limitações quanto à capacidade de leitura dos arquivos, exigindo que a análise seja elaborada em pequenos intervalos de tempo e depois, reagrupada.

6.7. Análise dos Protocolos de Roteamento

Nos últimos anos, muito se tem feito no sentido de melhorar o desempenho de protocolos de roteamento de redes *Ad Hoc,* tanto nas questões de performance, quanto nas questões de segurança. Os protocolos de roteamento desempenham um papel fundamental que influencia diretamente o desempenho de toda a rede, pois estas redes estão sujeitas à perda de comunicação devido à mobilidade dos nodos ou por interferências.

Na seção 4 foram apresentados os três grandes grupos de protocolos, pró-ativos, reativos e híbridos. Percebe-se que não há um consenso sobre qual é o melhor protocolo de roteamento, porque cada um contempla uma característica interessante, cada um possui suas vantagens e desvantagens. Dadas estas características, optou-se por realizar simulações com duas classes de protocolos, as pró-ativas e as reativas.

Protocolos	Não Seguros	Seguros
Pró-ativos	DSDV	SEAD
Reativos	DSR, AODV	Ariadne

TABELA 9 - PROTOCOLOS AVALIADOS

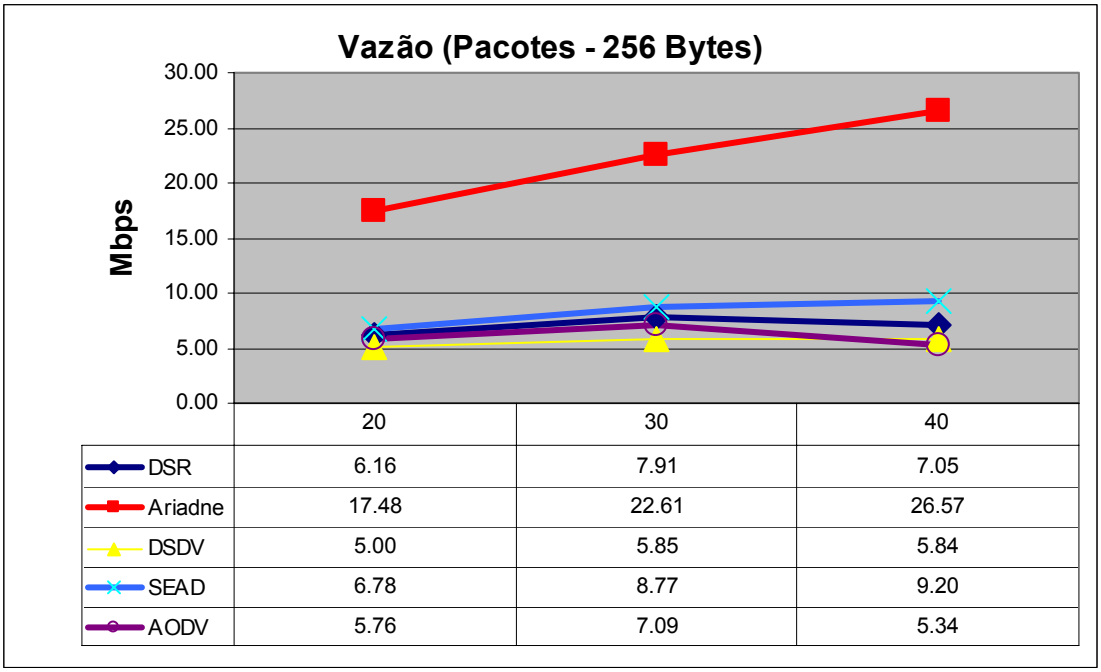
6.7.1. Vazão (*Throughput)*

A vazão ou througput é a taxa em bits, por segundos, de pacotes de dados recebidos por um nó. Pacotes excluídos, bem como pacotes de roteamento e de sinalização, não são levados em consideração. Essa métrica reflete a efetividade do protocolo. Através dos gráficos (1 e 2) é possível avaliar a média da vazão obtida por cada estação ao longo do tempo da simulação. Todas as estações estão programadas para iniciar as transmissões dos quadros de dados em t(0) e prosseguirem até que o tempo de parada da simulação seja atingido. Para este experimento, utilizamos o tempo de parada igual a 600 segundos.

Avaliando os resultados obtidos, pudemos observar que o protocolo Ariadne apresentou uma diferença significativa em todos os tamanhos de redes testados, independente dos tamanhos de pacotes utilizados. O protocolo SEAD também demonstrou um desempenho ligeiramente superior aos demais protocolos, sendo que, quanto testado em uma rede com 30

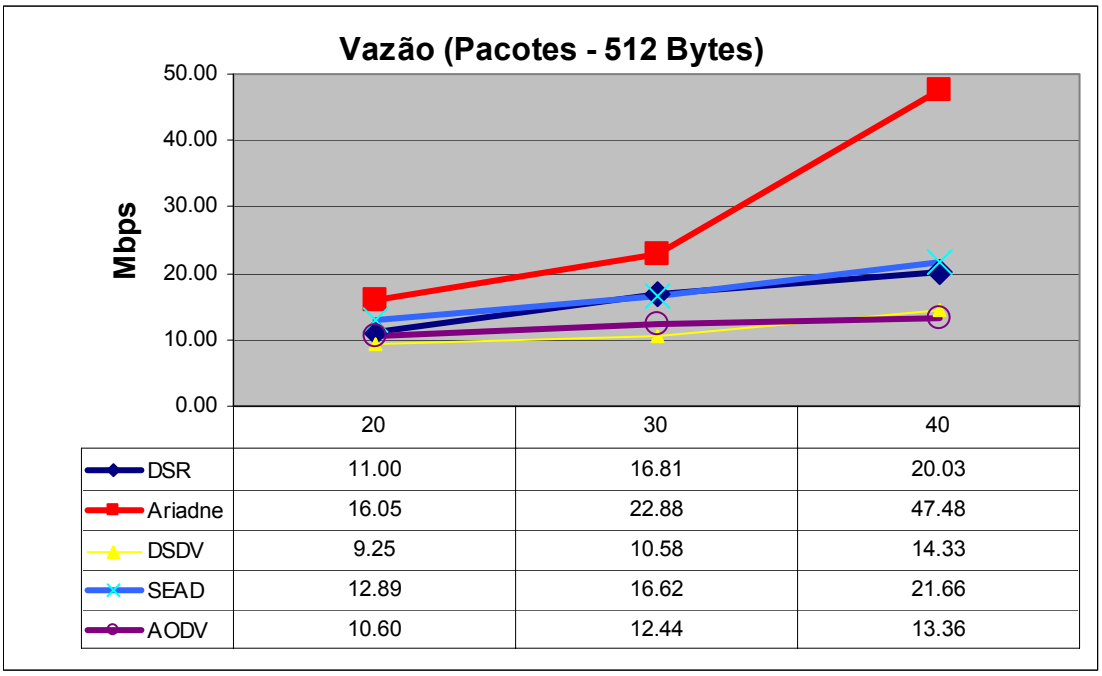
nodos e pacotes de 512 bytes, apresentou uma sensível defasagem, se comparado ao protocolo DSR. Como os protocolos DSR e SEAD são protocolos com classes de roteamento diferentes, onde o protocolo SEAD trabalha de forma pró-ativa é possível que, em uma situação onde as rotas entre os nodos estejam em suas tabelas de roteamento, este protocolo possa apresentar performance superior ao DSR, em um mesmo cenário.

Nos experimentos realizados, os gráficos 1 e 2 ilustram que, para uma fonte CBR, utilizando pacotes de 256 e 512 bytes, a maior vazão alcançada foi de 47,48 Mbps, o que evidencia que a vazão máxima obtida é dependente do tamanho dos pacotes adotados.



	20	30	40
DSR	6.16	7.91	7.05
Ariadne	17.48	22.61	26.57
DSDV	5.00	5.85	5.84
SEAD	6.78	8.77	9.20
AODV	5.76	7.09	5.34

Gráfico 1 – Vazão (througput) x Protocolos de Roteamento (pacotes de 256 bytes)



	20	30	40
DSR	11.00	16.81	20.03
Ariadne	16.05	22.88	47.48
DSDV	9.25	10.58	14.33
SEAD	12.89	16.62	21.66
AODV	10.60	12.44	13.36

Gráfico 2 – Vazão (througput) x Protocolos de Roteamento (pacotes de 512 bytes)

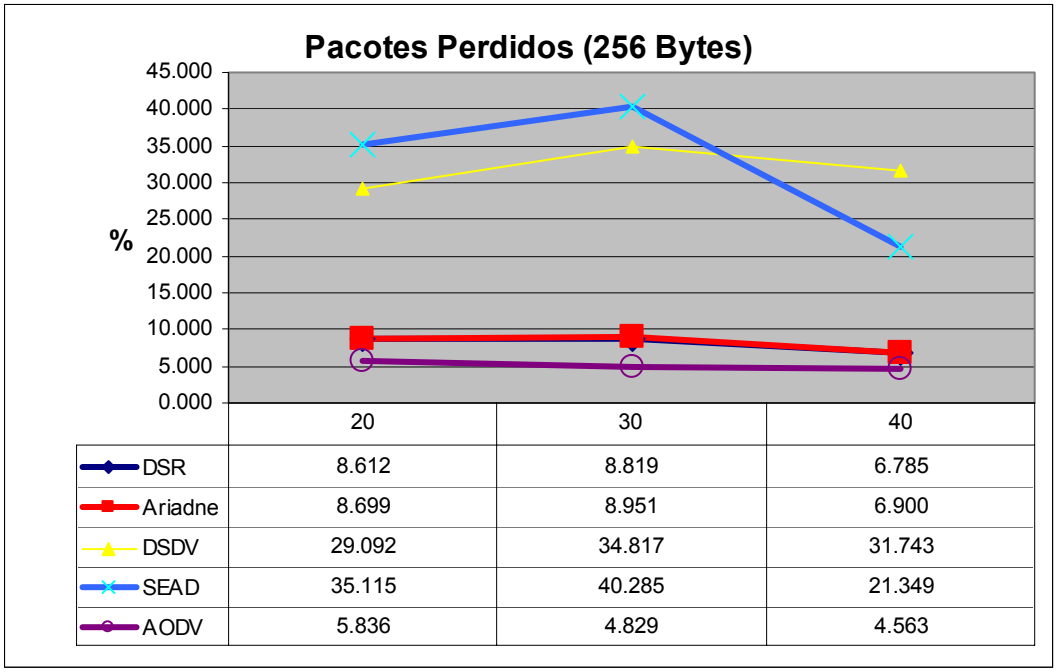
6.7.2. Taxa de Perda

As taxas de perdas de pacotes em um único sentido são calculadas no lado do receptor como a razão entre a quantidade de pacotes descartados pela quantidade de pacotes transmitidos.

Em redes *Wireless,* identificar a causa porque os pacotes são perdidos é uma tarefa difícil. Entretanto, nesta pesquisa identificamos as principais causas de perdas de pacotes: os pacotes eram perdidos devido ao congestionamento, ao tamanho dos pacotes, à capacidade da fila e à mobilidade das estações. Nas redes com 40 nodos percebeu-se que a perda de pacotes teve uma sensível redução. Entretanto, para as redes de 30 nodos a perda foi maior ao alcançado em redes com 20 e 40 nodos. Analisando o motivo deste comportamento, pode-se identificar que o padrão de mobilidade gerado para a simulação fez com que os nodos estivessem por demais concentrados e o número de nodos comunicando-se entre si fez com que o meio ficasse congestionado, gerando assim uma perda maior de pacotes.

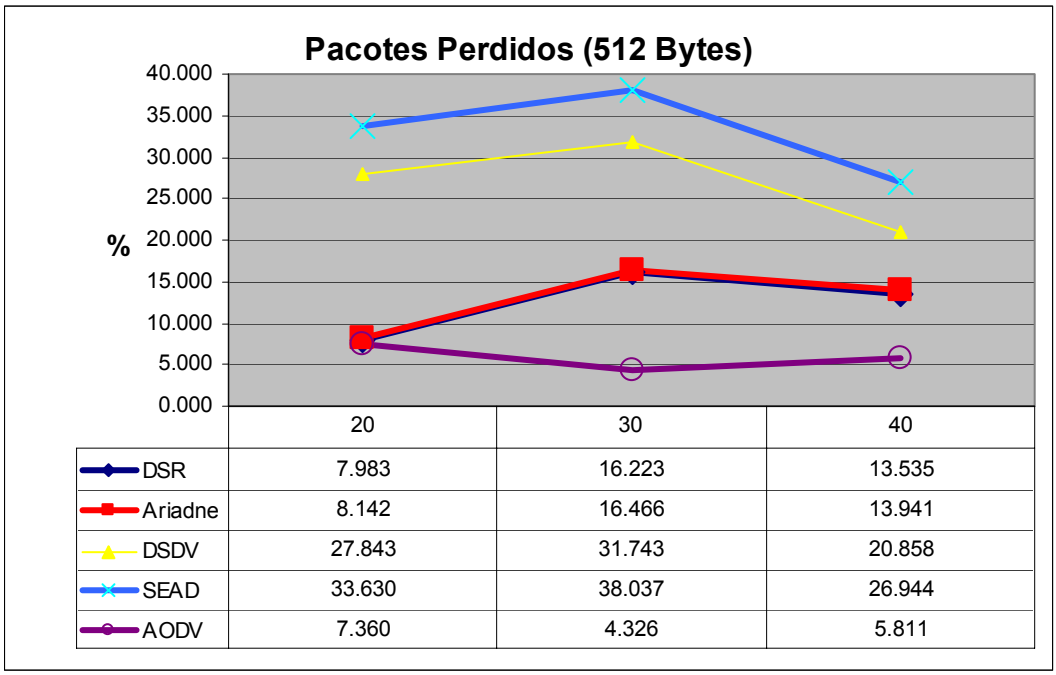
Os gráficos (3 e 4), dispostos a seguir, ilustram a taxa média de descarte de pacotes.

O protocolo AODV apresentou o melhor desempenho, sendo que o protocolo SEAD apresentou uma significativa perda com relação ao seu antecessor, exceto para redes com 40 nodos, utilizando-se de pacotes de 256 bytes, tendento a ter um desempenho melhor para redes maiores. Entretanto, não é possível afirmar que as implementações para agregar segurança ao protocolo são responsáveis por este desempenho, visto que, para as demais configurações de rede, os protocolos mostraram-se semelhantes, como é o caso do Ariadne, quando comparado ao DSR.



	20	30	40
DSR	8.612	8.819	6.785
Ariadne	8.699	8.951	6.900
DSDV	29.092	34.817	31.743
SEAD	35.115	40.285	21.349
AODV	5.836	4.829	4.563

Gráfico 3 – Taxa média de pacotes perdidos x Protocolos de Roteamento (pacotes de 256 bytes)

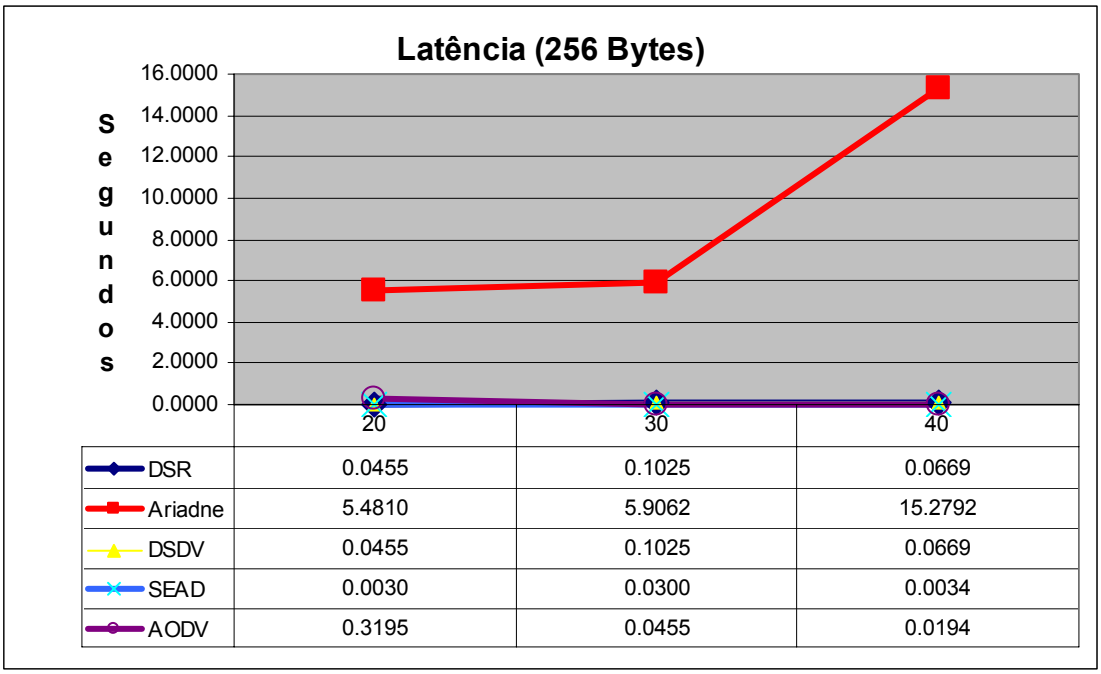


	20	30	40
DSR	7.983	16.223	13.535
Ariadne	8.142	16.466	13.941
DSDV	27.843	31.743	20.858
SEAD	33.630	38.037	26.944
AODV	7.360	4.326	5.811

Gráfico 4 – Taxa média de pacotes perdidos x Protocolos de Roteamento (pacotes de 512 bytes).

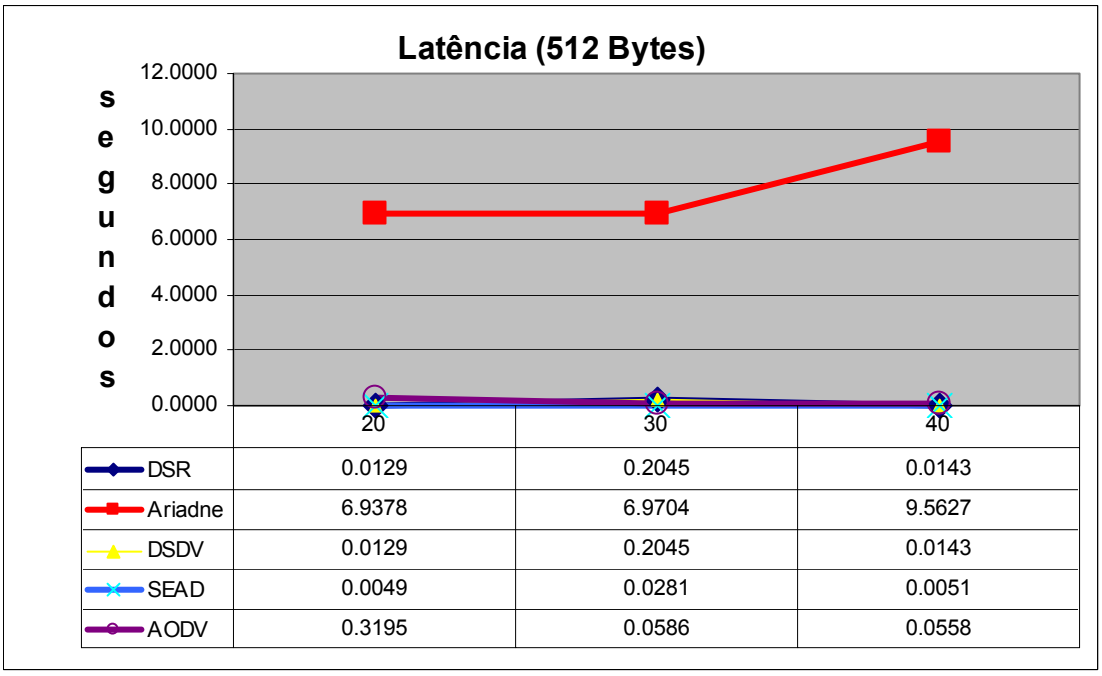
6.7.3. Latência

Á medida que a rede cresce, o número de nodos aumenta> Consequetemente, o número de transmições simultaneamente também aumenta, cresce a probabilidade de um nodo encontrar o meio ocupado, acarretando um aumento do tamanho da janela de contenção das estações e, conseqüentemente, contribuindo para o acréscimo dos tempos computados para a latência. Nesta análise, percebemos que o protocolo SEAD apresentou um desempenho sensivelmente melhor que sua versão não segura, o protocolo DSDV. O mesmo comportamento foi observado, considerando os dois tamanhos de pacotes simulados. Já os protocolos DSR e Ariadne apresentaram comportamentos muito diferentes, sendo que a versão não segura, o DSR, apresentou uma latência inferior ao protocolo Ariadne. Entretanto, não há evidência deste fato estar ligado aos tratamentos de segurança empregados ao Ariadne.



	20	30	40
DSR	0.0455	0.1025	0.0669
Ariadne	5.4810	5.9062	15.2792
DSDV	0.0455	0.1025	0.0669
SEAD	0.0030	0.0300	0.0034
AODV	0.3195	0.0455	0.0194

Gráfico 5 – Latência x Protocolos de Roteamento (pacotes de 256 bytes)

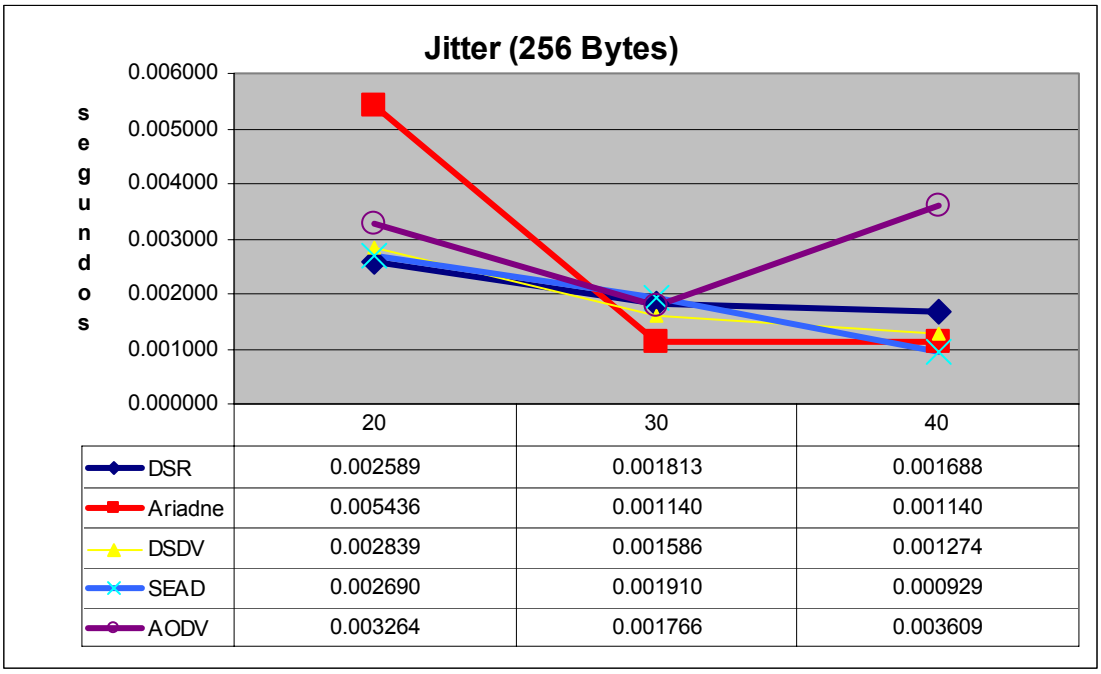


	20	30	40
DSR	0.0129	0.2045	0.0143
Ariadne	6.9378	6.9704	9.5627
DSDV	0.0129	0.2045	0.0143
SEAD	0.0049	0.0281	0.0051
AODV	0.3195	0.0586	0.0558

Gráfico 6– Latência x Protocolos de Roteamento (pacotes de 512 bytes)

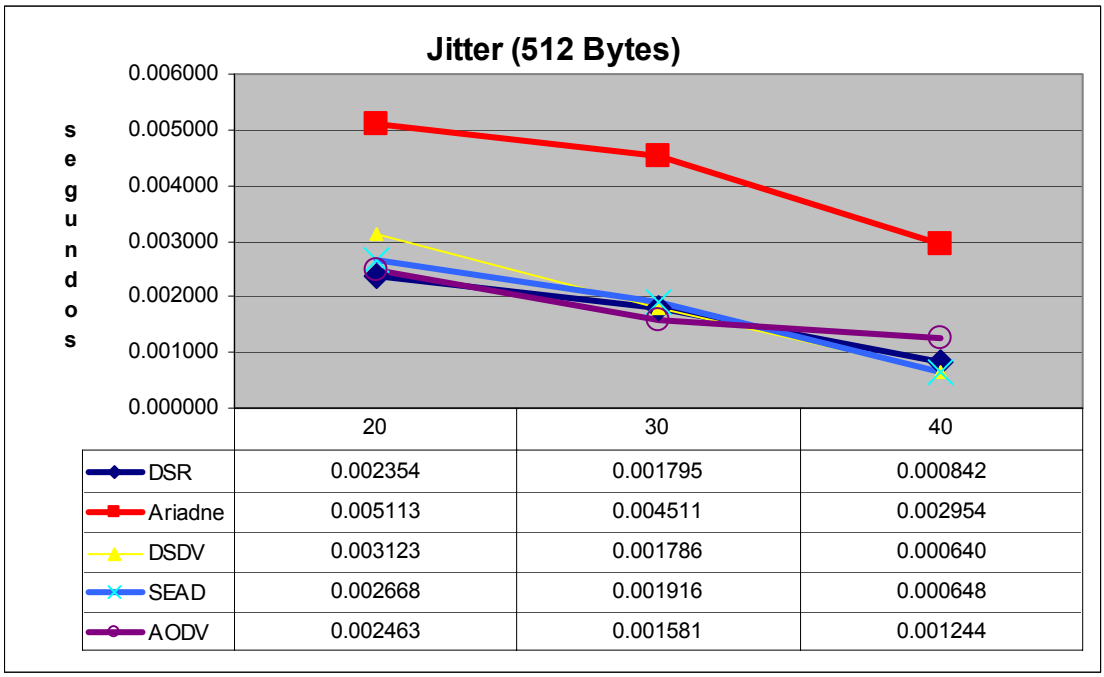
6.7.4. Jitter

Através da análise dos gráficos (7 e 8), percebe-se que a variação de retardo ocorrida variou bastante para todos os protocolos testados, sendo que o protocolo DSR apresentou a menor variação de retardo em uma rede com 20 nodos. Para redes com 30 nodos, utilizando pacotes de 256 bytes, a variação entre os protocolo foi muito pequena, apresentando um ligeiro destaque ao protocolo Ariadne. As redes maiores, com pacotes maiores, apresentaram uma melhora de desempenho. O fato se dá ao aumento de rotas disponíveis para o escoamento do tráfego da rede.



	20	30	40
DSR	0.002589	0.001813	0.001688
Ariadne	0.005436	0.001140	0.001140
DSDV	0.002839	0.001586	0.001274
SEAD	0.002690	0.001910	0.000929
AODV	0.003264	0.001766	0.003609

Gráfico 7– Jitter x Protocolos de Roteamento (pacotes de 256 bytes)



	20	30	40
DSR	0.002354	0.001795	0.000842
Ariadne	0.005113	0.004511	0.002954
DSDV	0.003123	0.001786	0.000640
SEAD	0.002668	0.001916	0.000648
AODV	0.002463	0.001581	0.001244

Gráfico 8– Jitter x Protocolos de Roteamento (pacotes de 512 bytes)

7. Conclusões e Trabalhos Futuros

A completa ausência de infra-estrutura e de pontos de acesso fixo contribue para que as comunicações se tornem mais difíceis. Provavelmente isto ocorre porque a topologia de rede é constantemente modificada, impossibilitando a pré-definição das rotas. A fim de melhor investigar estas quetões, neste trabalho, procedemos a comparação com cinco dos protocolos propostos para essas redes móveis.

É certo que a análise comparativa de protocolos de roteamento para MANET já foi realizada em várias pesquisas. Porém, os resultados aqui apresentados não são diretamente comparáveis aos resultados de nenhum trabalho anterior. Utilizamos um conjunto de protocolos diferentes, com mobilidade, tráfego e parâmetros de tamanho de rede diferentes e sobre um prisma diferente, que nesta pesquisa objetivou o impacto da agregação de segurança aos protocolos.

O que se observou é que não existe resposta para qual seria o melhor protocolo, tendo em vista que o desempenho de cada protocolo depende das condições e do ambiente de rede. Considerando mais relevantes as questões de segurança, os protocolos SEAD e Ariadne demonstraram ser as melhores opções. Entretanto, deve-se observar que a escolha do melhor protocolo estará diretamente ligada à sua aplicação.

Sobre as perspectivas de desempenho, pode-se perceber que o protocolo Ariadne, que implementa segurança, apresentou uma vazão maior. Já o protocolo SEAD apresentou uma performance muito semelhante à encontra em seu predecessor, o DSDV. Contudo, esta diferença pode estar relacionada ao fato do Ariadne ser um protocolo reativo. A mobilidade empregada neste experimento também pode ser um fator responsável por este resultado. Considerando os resultados apresentados nos experimentos desta dissertação, tanto o protocolo Ariadne quanto o SEAD são duas boas opções de protocolos de roteamento.

Percebeu-se, durante as simulações, utilizando as mesmas configurações da rede, tamanho de pacotes, número de nodos e taxa de transmissão, que, ao mudar o padrão de mobilidade e o cenário onde estavam distribuídos os nodos da rede, pode-se ter diferentes resultados. Este fato foi percebido ao gerarmos diferentes cenários, aos quais os protocolos foram submetidos.

No decorrer da pesquisa, pode-se perceber que o ambiente ideal para teste é um cenário real, ou seja, ter vários dispositivos *wireless* comunicando-se simultaneamente em áreas abertas e/ou edifícios similares a estruturas empresariais, pois, neste caso poderiam ser simulados ataques a fim de validar os aspectos de segurança e o comportamento das redes sobre condições reais. O simulador NS não nos permite efetuar simulações de ataques e nem tão pouco montar as estruturas empresariais encontradas nos cenários reais, e isto dificultou a apresentação de um resultado mais apurado, sob o aspecto de segurança dos protocolos testados. Outro fator que dificultou a realização desta dissertação foi à ferramenta Tracegraph, que realiza a análise dos traces produzidos pelas simulações. Esta ferramenta faz a extração aleatória de alguns eventos ocorridos e registrados nos arquivos de trace, atingindo um teto máximo de dez mil eventos, sob os quais realiza a avaliação das métricas de desempenho. Esta pequena quantidade de eventos pode ser pouco significativa, se considerarmos que estamos trabalhando com arquivos com mais de novecentas mil linhas. A situação ideal seria analisar os arquivos de trace por completo, sem exclusão de nenhum evento.

A continuidade desta dissertação poderia ser realizada através de uma análise mais aprofundada de outros protocolos de roteamento seguros, em ambientes reais. Além disso, dever-se-ia implementar a mesma análise sobre um simulador que contemplasse a simulação de ataques à rede. Avaliar a escalabilidade dos protocolos é também um assunto a ser investigado e, com isso também surgiria a oportunidade de trabalhar em uma ferramenta de análise dos grandes arquivos de trace, gerados pelo simulador.

Bibliografia

[Abbas, 2004] Abbas, Cláudia Jacy Barenco and Mendonça, Bruno José and Castro, Raoní Timo de, A Simulation-based Performance Analysis of Dynamic Routing Protocols for Mobile Ad Hoc Networks, Embedded and Ubiquitous Computing 2004, Japan, August 2004

[Aggelou and Tafazolli, 1999] G. Aggelou and R. Tafazolli, "RDMAR: A bandwidth-efficient routing protocol for mobile *Ad Hoc* networks," in ACM International Workshop on *Wireless* Mobile Multimedia (WoWMoM), August 1999.

[Amorim, 2002] Amorim, Glauco Fiorott. Análise de Desempenho de Protocolos de Roteamento com Diferenciação de Serviços em Redes de Comunicação Móvel *Ad Hoc* / Glauco Fiorott Amorim. -- Rio de Janeiro : Instituto Militar de Engenharia, 2002.

[Bantz, 1994] BANTZ, DF and BANCHOT, FJ Wireles LAN Design Alternatives. IEEE Network Magazine. Vol.8. March/April 1994.

[Basagni, 1998] S. Basagni, I. Chlamtac, V.R. Syrotivk, B.A. Woodward, A distance e.ect algorithm for mobility (DREAM), in: Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom_98), Dallas, TX, 1998.

[Beard, 2002] D. Beard and Yi Yang "Known Threats to Routing Protocols" draft-beard-rpsec-routing-threats-00.txt, RFC2026, Outubro, 2002

[Bhargav, 2001] TBRPF (Topology Broadcast based on Reverse-Path Forwarding routing protocol) - BHARGAV BELLUR, RICHARD G. OGIER, FRED L. TEMPLIN Topology Broadcast Based on Reverse-Path Forwarding (TBRPF) Internet Draft, draft-ietf-manet-tbrpf-01.txt, work in progress, June 2001

[Bradley , 1997] Bradley R. Smith, Shree Murthy, and J.J. Garcia-Luna-Aceves. "Securing Distance Vector Routing Protocols," Proceedings of the 1997 Symposium on Network and Distributed Systems Security (NDSS '97), pages 85–92, February 1997.

[Câmara, 1999] Câmara, Daniel e Loureiro, Antonio A. F., Curso de Redes de Computação Móvel *Ad Hoc*. Jornada de Atualização em Informática, Rio de Janeiro, 1999.

[Câmara, 2001] Daniel Câmara, Roteamento em Redes *Ad Hoc*, Dissertação de Mestrado. UFMG. Junho, 2001

[Cheng, 1998] CHENG, T.-W. e GERLA, Mário. Global State Routing: A new routing scheme for *Ad Hoc* wireless networks. In Proceedings of the IEEE International Conference on Communications (ICC'98), Atlanta, June 1998.

[Chiang, 1997] C.-C. Chiang, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel" Proc. IEEE SICON'97, Apr.1997, pp.197-211. http://www.ics.uci.edu/~atm/adhoc/paper-collection/gerla-routing-clustered-sicon97.pdf

[Corson And Ephremides, 1995] LMR (Lightweight Mobile Routing protocol) - M.S. CORSON AND A. EPHREMIDES Lightweight Mobile Routing protocol (LMR), A distributed routing algorithm for mobile wireless networks, Wireless Networks 1 (1995).

[Corson, 1999] Corson, S. e Macker, J. Mobile *Ad Hoc* networks (manet): Routing protocol performance issues and evaluation considerations. IETF RFC 2501, January 1999.

[Dahill, 2001] B. Dahill, B. N. Levine, E. Royer, and C. Shields. A Secure Routing Protocol for *Ad Hoc* Networks. 2001.

[Dube And Tripathi, 1997] R.Dube, C.D. Rais, K. Wang and S.K. Tripathi, "Signal Stability Based Adaptive Routing for Ad-hoc mobile networks", IEEE Personal Communications, Feb. 1997.

[Garcia, 1999] J.J. Garcia-Luna-Aceves, C. Marcelo Spohn, Source-tree routing in wireless networks, draft-ietf-manet-star-00.txt, RFC2026, Outubro, 1999.

[Geier, 1996] GEIER, Jim. Wireless Networking Handbook. Indianapolis, Estados Unidos: New Riders Publishing, 1996, 413p.

[Gerla, 2002] Mario Gerla, Fisheye state routing protocol (FSR) for *Ad Hoc* networks, Internet Draft, draft-ietf-manet-aodv-03.txt, work in progress, 2002.

[Günes, 2002] - Mesut Günes., (Ant-based Routing Algorithm for Mobile Ad-Hoc Networks) - ARA - the ant-colony based routing algorithm for manets, In Stephan Olariu, editor, Proceedings of the 2002 ICPP Workshop on *Ad Hoc* Networks (IWAHN 2002), pages 79-85, IEEE Computer Society Press, August 2002

[Hass, 1997] Hass, Zygmunt J. A new Routing Protocol for the Reconfigurable Wireless Networks. IEEE ICUPC'97, San Diego, October 1997.

[Haas , 1998] Hass, J. Z. e Pearlman, M.R. *The zone routing protocol (ZRP) for Ad Hoc networks.* Internet Draft August 1998.

[Hayes, 1996] HAYES, Vic. Tutorial on 802.11 to 802. Documento IEEE P802.11-96/49A, disponível por FTP anônimo em atg.apple.com no diretório /pub/802.11/ibmpc/1996_docs, Março de 1996.

[Horstmann, 2002] HORSTMANN, G. Avaliação de Esquemas de Fila para o Host Controller Interface do Bluetooth. Dissertação de Mestrado. UFSC. Outubro, 2002.

[Hu, 2002] Yih-Chun Hu, D. B. Johnson, and A. Perrig. SEAD: Secure efficient distance vector routing for mobile wireless *Ad Hoc* networks. In Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications, June 2002.

[Hu, 2002b] Yih-Chun Hu, Adrian Perrig and David B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Wireless *Ad Hoc* Networks" Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (MobiCom 2002), pp. 12-23, ACM, Atlanta, GA, September 2002.

[Hu, 2003] Yih-Chun Hu, A. Perrig, D.B. Johnson, Packet leashes: a defense against wormhole attacks in wireless networks, In Proceedings of IEEE Infocom, April 2003.

[Hu and Perring and Johnson, 2002] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. "Rushing Attacks and Defense in Wireless *Ad Hoc* Network Routing Protocols". Technical Report TR01- 384, Department of Computer Science, Rice University, June 2002.

[Hu, 2003b] Yih-Chun Hu, A. Perrig, and D. B. Johnson. Efficient Security Mechanisms for Routing Protocols. In Proceedings of NDSS, February 2003.

[Hu, 2003c] Yih-Chun. Hu, A. Perrig, D.B. Johnson, Packet leashes: a defense against wormhole attacks in wireless networks, in: IEEE Infocom, 2003.

[IEEE, 1999] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. Wireless LAN Medium Access Control (MAC) and Physical (PHY) Specifications. IEEE Standard 802.11, 1999.

[IETF, 1999] Internet Engineering Task Force. http://www.ietf.org.

[IETF 1999] Internet Engineering Task Force. http://www.ietf.org./ [MANET99] Mobile *Ad Hoc* Networks (manet). http://www.ietf.org/html.charters/manet-charter.html.

[ISO, 1989] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. Página da World Wide Web. url: http://www.iso.org/. Acesso em: Janeiro de 2003.

[Jacquet and Mühlenthaler, 2001] Philippe Jacquet and Paul Mühlenthaler. Optimized link state routing protocol for *Ad Hoc* networks. In Proceedings of the IEEE International Multi-Topic Conference (INMIC), Pakistan, June 2001.

[Jacquet and Mühlenthaler, 2002] Philippe Jacquet, Paul Mühlenthaler, and Amir Qayyum. Optimized link state routing protocol. Internet Draft, draft-ietf-manet-olsr-07.txt, Work-in-progress, December 2002.

[Jimenez and Shrivastava, 2004] Carlos Molina-Jimenez, Santosh Shrivastava, Jon Crowcroft and Panos Gevros, On the Monitoring of Contractual Service Level Agreements, School of Computing Science, University of Newcastle upon Tyne, UK. First IEEE International Workshop on Electronic Contracting (WEC'04), Julho de 2004.

[Joa-ng, 1999] Mario Joa-Ng, I.-T. Lu, A peer-to-peer zone-based two-level link state routing for mobile *Ad Hoc* networks, IEEE Journal on Selected Areas in Communications, Vol. 17, Agosto de 1999, páginas 1415–1425.

[Johnson and Maltz, 1996] David B. Johnson and David A. Maltz. "Dynamic Source Routing in *Ad Hoc* Wireless Networks", in Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5, pages 153-181, Kluwer Academic Publishers, 1996.

[Johnston and Sewry, 1999] ROBERT H.B. JOHNSTON AND DAVID A. SEWRY, A MODEL FOR THE DEVELOPMENT OF SERVICE AGREEMENTS IN THE INFORMATION AND COMMUNICATION TECHNOLOGY SECTOR, July, 1999, Disponível em http://www.abode.za.net/Saicsit.pdf, acessado em 04/12/2004.

[Katz, 1994] Katz, R.H. "Adaptation and Mobility in Wireless Information Systens". IEEE Personal Communications Magazine. Vol 1. No 1. 1994.

[Kumar, 2001] Brijesh Kumar. “Integration of Security in Network Routing Protocols”. SIGSAC Review, 11(2):18–25, 1993. (CCS-8), pages 28–37, November 2001.

[Lamia 2003] Lamia, R.; Qiang, N.; Thierry, T. Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks. Wireless Communications and Network Conference (WCNC2003). New Orleans, USA, March, 2003.

[Manet, 1999] Mobile Ad Hoc Networks (manet). http://www.ietf.org/html.charters/manet-charter.html.

[Malek, 2005] Malek, Jaroslaw. Trace Graph. Disponível em : < http://www. geocities.com/ tracegraph/, site official da ferramenta, acessado em 03 de Janeiro de 2005.

[Michiardi and Molva, 2003] Pietro Michiardi and Refik Molva. *Ad Hoc* Networks Security. In ST Microelectronics Journal of System Research, to appear in 2003.

[Mingliang, 1999] Mingliang Jiang, J. Ji, Y.C. Tay, Cluster based routing protocol, Internet Draft, draft-ietf-manet-cbrp-spec-01.txt, work inprogress, 1999.

[Molva and Michiardi, 2002] Refik Molva and Pietro Michiardi, Security in *Ad Hoc* Networks – Citeseer – scientific literature digital library.

[Monarch, 2004] MONARCH PROJECT. Wireless and Mobility Extensions to ns-2. Página da World Wide Web. url: http://www.monarch.cs.cmu.edu/cmu-ns.html. Acesso em Outubro de 2004.

[Murthy, 1996] S. Murthy and J.J. Garcia-Luna-Aceves, “An efficient routing protocol for wireless networks,” In *ACM Mobile Networks and Applications Journal*, October 1996, 183-197.

[Naldurg, 2001] S. Yi, P. Naldurg, and R. Kravets. Security-Aware Ad-Hoc Routing for Wireless Networks. August 2001.

[Network Simulator, 2004a] Network Simulator (versão 2.27). Página da World Wide Web. url: http://www.isi.edu/nsnam/ns/ns-documentation.html. Acesso em: Janeiro de 2004.

[Network Simulator, 2004] Network Simulator (versão 2.27). Página da World Wide Web. url: http://www.isi.edu/nsnam/ns/. Acesso em: Janeiro de 2004.

[Nikaein and Laboid, 2002] Navid Nikaein, Houda Laboid, Christian Bonnet, Distributed dynamic routing algorithm (ddr) for mobile *Ad Hoc* networks, in: Proceedings of the MobiHOC 2000: First Annual Workshop on Mobile *Ad Hoc* Networking and Computing, 2000.

[Nikaein and Bonnet, 2001] Navid Nikaein, Christian Bonnet, Neda Nikaein, Harp-hybrid *Ad Hoc* routing protocol, in: Proceedings of IST: International Symposium on Telecommunications, September 1–3 Tehran, Iran, 2001.

[Pei, 1999] G. Pei, M. Gerla, X. Hong, C. Chiang, A wireless hierarchical routing protocol with group mobility, in: Proceedings of Wireless Communications and Networking, New Orleans, 1999.

[Perkins, 2001] C.E. Perkins, *Ad Hoc* Networking, Addison-Wesley, (ISBN: 0201309769), 2001.

[Perkins, 2003] Perkins, C. E.; Belding-Royer, E. M.; Das, S. R.; "*Ad Hoc* On-Demand Distance Vector Routing", Request for Comments: 3561, rfc3561.txt, julho de 2003.

[Portnoi e Araújo, 2002] Portnoi, Marcos, Araújo, Rafael Gonçalves Bezerra de Araújo, Visão Geral da Ferramenta de Simulação de Redes, 2002. Disponível em: http://locksmith.orcishweb.com/articles/networksimulator-sepa.pdf

[Raju, 1999] J. Raju, J. Garcia-Luna-Aceves, A new approach to ondemand loop-free multipath routing, in: Proceedings of the 8th Annual IEEE International Conference on Computer Communications and Networks (ICCCN), Boston, MA, October 1999, pp. 522–527.

[Ramos, 1998] Ramos, C. e Rochol, J. "Análise de Desempenho por Simulação de Subcamada MAC do Padrão IEEE 802.11 para Redes Locais sem Fio". Anais do XVI Simpósio Brasileiro de Redes de Computadores. Universidade Federal Fluminense. Rio de Janeiro, 25- 28 de maio de 1998.,

[Ribas, 2002] RIBAS, Júlio César da Costa. Perfil de Link Sem Fio em Ambiente Aberto: Avaliação através de Medições. Dissertação de Mestrado. UFSC. Julho, 2002.

[Royer and Toh, 1999] Royer, E. M., e Toh, C., "A Review of Current Routing Protocols for *Ad Hoc* Mobile Wireless Networks",IEEE Personal Communications, pp. 46-55, abril de 1999.

[Sanzgiri, 2002] Sanzgiri, K., Dahill, B., Levine, B. N., Shields, C., and Belding-Royer, E. M. (2002). A secure routing protocol for *Ad Hoc* networks.

[Santos, 1999] SANTOS, M. C. M., AMORIM, G. F., SOBRAL, A. A. Roteamento em Redes de Comunicação Sem Fio - *Ad Hoc*. Revista da Marinha - Casnav. , v.1, 1999. Referências adicionais : Brasil/Português. Meio de divulgação: Meio digital, Home page: http://www.gta.ufrj.br/~myrna/ http://www.gta.ufrj.br/~glauco/Spolm99_RoteamentoRedesComunicacaoSemFioAdHoc.htm

[Selea and Moldoveanu, 2004] Radu Selea, Bogdan Moldoveanu, Operation Support System Interface Specification for 802.16 fixed Wireless Systems, Project IEEE 802.16 Broadband Wireless Access Working Group - Agosto de 2004.

[Shirey, 2000] R.Shirey, Internet Security Glossary, RFC 2828, May 2000

[Silva, 2004] SILVA, Madalena Pereira da. Perfil de Link Sem Fio em Ambiente Aberto Análise de Desempenho e Diferenciação na Camada MAC do Padrão IEEE 802.11. Dissertação de Mestrado. UFSC. Janeiro, 2004.

[Skene and Emmerich, 2004] James Skene, D. Davide Lamanna, Wolfgang Emmerich, “Precise Service Level Agreements”, Department of Computer Science, University College London, Proc. 26th Int’l Conf. On Software Engineering (ICSE'04), May 2004,

[Sturm, 2000] Sturm, Rick; Wayne, Morris; Jander, Mary. **Foundations of Service Level Management**. Editora SAMS. 2000.

[Su and Gerla, 1999] Willian Su and Mario Gerla, ìIPv6 Flow Handoff in Ad-Hoc Wireless Networks Using Mobility Prediction,î Proceedings of IEEE GLOBECOM' 99, Rio de Janeiro, Brazil, Dec. 1999, pp. 271-275.

[Tanenbaum, 1996] Tanenbaum, Andrew S., Computer Networks. Third Edition – Prentice Hall PTR, Ney Jersey, 1996.

[UCLA, 2004] UCLA Wireless Adaptative Mobility Laboratory. [on line]. Disponível em: http://www.cs.ucla.edu/NRL/wireless/ [capturado em em 21/04/2004].

[Wamis, 2004] WAMIS, UCLA Wireless Adaptive Mobile Information System. [on line]. Disponível em: http://www.lk.cs.ucla.edu/wamis.html [capturado em 21/04/2004].

[Wang, 2002] Wang,W. and Bhargava, B. (2002). On vulnerability and protection of *Ad Hoc* on-demand distance vector protocol. Technical report, Technical report, TR-2002-18, CERIAS Security Research Center, Purdue University.

[Wang, 2002B] Wang, W., Lu, Y., and Bhargava, B. (2002). On security study of two distance-vector routing protocols for mobile *Ad Hoc* networks. Technical report, Technical report, Dept.
of Computer Sciences, Purdue University.

[Wang, 2003] Weichao Wang, Yi Lu, Bharat K. Bhargava "On Vulnerability and Protection of *Ad Hoc* On-demand Distance Vector Protocol" Proceedings of International Conference on Telecommunication (ICT), France, February 2003

[Wustenhoff, 2002] WUSTENHOFF, EDWARD , Service Level Agreement in the Data Center, Sun BluePrints™ OnLine - April 2002, Disponível em http://www.sun.com/blueprints/0402/sla.pdf, acessado em Outubro de 2004.

[Yacoub, 1993] YACOUB, Michel D., Foundations of Mobile Radio Engineering. Florida: CRC Press, 1993. 473 p. ISBN 0-8493-8677-2

[Zhang, 2000] Y. Zhang and W. Lee. Intrusion Detection in Wireless *Ad Hoc* Networks. MobiCom'2000, Agosto 2000.

[Young-Bae, 1998] Ko, Young-Bae e Vaidya, Nitin H. Location Aided Routing (LAR) in mobile *Ad Hoc* networks. In Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, pages 66-75, Dallas, Texas, USA, October 1998.

[Zapata, 2001] Zapata,Manel Guerrero Secure *Ad Hoc* On-Demand Distance Vector (SAODV) Routing, CiteSeer Scientific Literature Digital Library, Outubro/2001

[Zhou, 2004] Zhou, Nianjun, Routing Overhead In Variable Topology Networks, Tese submetida a defesa - Rensselaer Polytechnic Institute, Troy, New York – Dezembro 2004.

[Royer and Toh, 1999] Royer, Elizabeth M., Toh, Chai-Keong, A Review of Current Routing Protocols for *Ad Hoc* Mobile Wireless Networks, IEEE Personal Communications, Vol. 6, No. 2, pp. 46-55, April 1999.

[Portnoi, 2002] Portnoi, Marcos, Network Simulator – Visão Geral da Ferramenta de Simulação de Redes, 2002.

[Yoon and Liu and Noble, 2003] Yoon, Jungkeun and Liu, Mingyan and Noble, Brian, Random Waypoint Considered Harmful, In Proceedings of INFOCOM. IEEE, 2003.